Appendix B. Software Libraries

Some libraries have both static and shared versions. The shared libraries will be indicated by "(shared)" in the heading. In all cases, the name of a shared library ends in "Sh". However, some static libraries also have names which end in "Sh".

The first sentence of a class description is taken from the comments, if any, in the code for that class.

library agentevent

/ecs/formal/CSS/DOF/src/LOGGING/agentlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- **EcUtAgentLogger**

library agenteventSh (shared)

/ecs/formal/CSS/DOF/src/LOGGING/agentlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- **EcUtAgentLogger**

library ClDtProfileToOdl

/ecs/formal/CLS/DESKT/src/OdlProfile

This library is used to read and write ODL parameters specific for the User Profile Gateway.

Classes

- **ClDtDataValue**
- **ClDtDateTime**
- **ClDtODLAggregate**
- **ClDtODLNonAggregate** -- This class performs extracting non aggregate informations from input aggregate ODL tree.
- **ClDtODLParameter**
- **ClDtODLTree**

- **ClDtRSTAggregate**
- **ClDtRSTNonAggregate**

library ClWbJtdb

/ecs/formal/CLS/WKBCH/ClWbMp/db

library ClWbJtFolioUtils

/ecs/formal/CLS/WKBCH/ClWbMp/foliod

library ClWbJtgd

/ecs/formal/CLS/WKBCH/ClWbMp/gd

library ClWbJtMap

/ecs/formal/CLS/WKBCH/ClWbMp/maplib

library ClWbJtProjections

/ecs/formal/CLS/WKBCH/ClWbMp/libproj

library Common

/ecs/formal/DM/GatewayCSCI/src/Common

This library contains classes for the EcsToV0Server and V0ToEcsServer executables. These classes are the framework used to build the executables.

Classes

- **DmGwConfigFile** -- Read a configuration item from the EcDmV0ToEcsGateway CFG file.
- **DmGwConfigItems** -- This class will get items from Process Framework that have been read from the configuration file by PF.
- **DmGwDataValue** -- This class performs store and get the data value.
- **DmGwDateTime** -- This class performs the set time, date and get time, date.
- **DmGwDistribution** -- This class performs get product history information.
- **DmGwECSCircle** -- This class performs get circle, radius, and center for the circle search.
- **DmGwECSPoint** -- This class performs get longitude and latitude for the point search.
- **DmGwECSPolygon --** This class performs get polygon for polygon search.
- **DmGwECSRectangle** -- This class performs get rectangle, get north, south longitude and latitude for the rectangle search.
- **DmGwEvent --** This class performs the logging events in Gateway managed server.
- **DmGwExtendedSearch** -- This class represents the V0 Extended Search group and consists functions to set and get the parameters for the specialized criteria group.

- **DmGwGlobalGranulesOnly** -- This class performs creation of object DmGwGlobalGranulesOnly.
- **DmGwImageStream** -- This class performs save image file, get length of image size, and get name of the image.
- **DmGwMediaFormat** -- This class performs get media format, get parameter list.
- **DmGwMediaInfo** -- This class performs get media type, get media format list.
- **DmGwMediaParameter** -- This class performs creating DmGwMediaParameter object.
- **DmGwODLAggregate** -- This class performs get inventory search, get directory search, get browser request, get product request, get quit, get abort, get acknowledge, and extract informations from the input ODL tree.
- **DmGwODLNonAggregate** -- This class performs extracting non aggregate informations from input aggregate ODL tree.
- **DmGwODLParameter** -- This class builds and gets parameters from ODL tree.
- **DmGwODLTree** -- This class builds/gets root node of an input ODL Tree.
- **DmGwPathRowLoc** -- This class represents the V0 Search group and consists functions to set and get the parameters for the Path Row Loc group.
- **DmGwPointLoc** -- This class represents the V0 point location group and consists functions to set and get the latitude and longitude of the point location.
- **DmGwPolygonLoc** -- This class represents V0 Polygon Loc group and consists of functions to get and set the attributes using ECS types.
- **DmGwRangeLoc** -- This class represents V0 RangeLoc group.
- **DmGwRSTAggregate** -- This class builds all necessary ODL aggregates for the output ODL tree.
- **DmGwRSTNonAggregate** -- This class builds the non aggregate elements of the ODL tree for response messages.
- **DmGwSpatial** -- Base class for all the spatial classes that are supported by the V0 gateway.
- **DmGwTemporal** -- This class constructs the temporal constraint, does conversion from V0 to ECS temporal constraint.
- **DmGwTemporalVec** -- This class supports the conversion of V0 temporal type which includes startday of the year and stop day of the year to multiple ECS date ranges.
- **DmGwV0Request** -- This is the base class for the request types that can be processed by V0 gateway.
- **StringVec** -- This class holds a lists of string pointers and forms a comma seperated list when required for SQL construction.

library CsEmMailRelA

/ecs/formal/CSS/DOF/src/EMAIL/email

This Library is used for the emails using sendmail utility and the nnpost for Bulletin board messages.

Classes

- **CsBBMailRelA** -- Implements CsBBMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic post messasges.
- **CsEmMailRelA** -- Implements CsEmMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic mail messasges.

library CsEmMailRelASh (shared)

/ecs/formal/CSS/DOF/src/EMAIL/email

This Library is used for the emails using sendmail utility and the nnpost for Bulletin board messages.

Classes

- **CsBBMailRelA** -- Implements CsBBMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic post messasges.
- **CsEmMailRelA** -- Implements CsEmMailRelA class for interactive(for operators) and development(API, for applications) interfaces to manage electronic mail messasges.

library CsFtFTPRelA

/ecs/formal/CSS/DOF/src/FTP/ftp

This library provides a Wrapper for FTP. User can use it for all the FTP needs, such as sending or receiving files to/from remote Hosts.

Classes

- **CsFtFTPRelA** -- By instantiating a CsFtFTPRelA object and sending the object messages, a program can perform programmed dialogue with a ftp process and effectively control it.

library CsFtFTPRelASh (shared)

/ecs/formal/CSS/DOF/src/FTP/ftp

This library provides a Wrapper for FTP. User can use it for all the FTP needs, such as sending or receiving files to/from remote Hosts.

Classes

- **CsFtFTPRelA** -- By instantiating a CsFtFTPRelA object and sending the object messages, a program can perform programmed dialogue with a ftp process and effectively control it.

library CsFtFTPSchedObj

/ecs/formal/CSS/DOF/src/FTPSCHED

This library provides functions to use FTP at a scheduled time by the User.

Classes

- **CsFtFileListObj**
- **CsFtFTPSchedObj** -- This class is designed to be created by the Managed Process Framework to allow developers to schedule file transfers at a given time.

library CsFtFTPSchedObjSh (shared)

/ecs/formal/CSS/DOF/src/FTPSCHED

This library provides functions to use FTP at a scheduled time by the User.

Classes

- **CsFtFileListObj**
- **CsFtFTPSchedObj** -- This class is designed to be created by the Managed Process Framework to allow developers to schedule file transfers at a given time.

library CsSeCryptoDes

/ecs/formal/CSS/DOF/src/CRYPTO

This library contains the APIs for the DES encryption algorithm. This library is primarily used for encrypting passwords for storage. The primary functions are DesEncrypt and DesDecrypt. Both return strings containing the encrypted/decrypted text. Two other functions, DesGetMsgLen and DesGetCypherLen are provided for the application programmer to obtain the length of the returned string since it may vary slightly due to the encryption/decryption algorithm.

Classes

- **CsSeCryptoDes** -- This class implements both standard and modified forms of the NBS Data Encryption Standard (DES).

library DdictTestMsg

/ecs/formal/DM/DDICT/src/ddtestmsg

Classes

- **deleteAcronymMsg**
- **deleteAttributeGroupMsg**
- **deleteBoundingRectangleMsg**
- **deleteCollKeywordMsg**

- **deleteCollMsg**
- **deleteCrossRefMsg**
- **deleteDataOrigMsg**
- **deleteEcsAttrMsg**
- **deleteEquivXrefMsg**
- **deleteGlossaryMsg**
- **deleteInfoMgrMsg**
- **deleteInstPlatXrefMsg**
- **deleteInstrumentXrefMsg**
- **deleteInstSensXrefMsg**
- **deleteKeywordAliasMsg**
- **deleteKeywordMsg**
- **deletePlatformXrefMsg**
- **deleteSensorXrefMsg**
- **deleteSpatialKeywordMsg**
- **deleteTemporalKeywordMsg**
- **insertAcronymMsg**
- **insertAttrGroupMsg**
- **insertAttrMsg**
- **insertBoundingRectangleMsg**
- **insertCollKeywordMsg**
- **insertCollMsg**
- **insertDataOrigMsg**
- **insertEcsAttrMsg**
- **insertEquivXrefMsg**
- **insertGlossaryMsg**
- **insertInfoMgrMsg**
- **insertInstPlatXrefMsg**
- **insertInstXrefMsg**
- **insertKeywordAliasMsg**
- **insertKeywordMsg**
- **insertPlatXrefMsg**
- **insertSensInstXrefMsg**

- **insertSensXrefMsg**
- **insertSpatialKeywordMsg**
- **insertTemporalKeywordMsg**
- **releaseCollectionMsg**
- **unReleaseCollectionMsg**
- **updateAcronymMsg**
- **updateAttrGroupMsg**
- **updateCollMsg**
- **updateDataOrigMsg**
- **updateEcsAttrMsg**
- **updateGlossaryMsg**
- **updateInfoMgrMsg**
- **updateKeywordMsg**

library des

/ecs/formal/CSS/DOF/src/CRYPTO

This library represents the core DES APIs. This library provides APIs for setting the key, initializing the algorithm and performing the transformations.

library DmAsGwAsterReqProc

/ecs/formal/DM/ASTGW/src/AsterReqProc

The lib_DmAsGwAsterReqProc library contains classes for processing ASTER to ECS requests. These classes convert an ODL file received from the ASTER client to a GlParameter list. The request is then processed and the results are converted back to ODL to be sent to the ASTER client.

Classes

- **DmAsGwAsterAdSubmit** -- This class aids in interfacing with the Advertising Service.
- **DmAsGwAsterBroRequest** -- This class processes an Aster to Ecs Browse request.
- **DmAsGwAsterBroRequestor** -- This class is used to submit a query to the SDSRV via the LIM libs.
- **DmAsGwAsterCancelRequest** -- The operations "concept" for canceling a product request is to first ask for product status, and obtain the top-level initiator request ID and then each sub-request ID.
- **DmAsGwAsterDirRequest** -- The managed server should perform a directory search request by invoking Initialize followed by the SendAndReceive function.
- **DmAsGwAsterInvRequest** -- Submit is called by the managed server and the output ODL tree is passed in by reference.

- **DmAsGwAsterInvRequestor** -- This class is used to submit a query to the SDSRV via the LIM libs.
- **DmAsGwAsterODLAgg** -- This class supports input ODL tree aggregates.
- **DmAsGwAsterODLNonAgg** -- This class implements Get~() functions that were not provided for in the base class DmGwODLNonAggregate.
- **DmAsGwAsterPriceEstRequest** -- This class extracts the PRICE_ESTIMATE_REQUEST group parameters specified in the constructor's ODL argument, converts it into a GlParameterList structure to submit to the LIMGR using the DmAsGwAsterPriceEstRequestor class. After the GlParameterList
- **DmAsGwAsterPriceEstRequestor** -- This is the delegatee for a given Price Estimate Request whose ODL request message was passed to its constructor.
- **DmAsGwAsterProductStatusRequest** -- The managed server should perform a product status request by invoking Initialize followed by the SendAndReceive function.
- **DmAsGwAsterProductUpdate** -- The operations concept for the product update is that when a previously ordered product is shipped to the client who made the original product request, a product update information message is sent and a product update acknowledge is returned as a .
- **DmAsGwAsterProRequest** -- This class extracts data from a Product request ODLTree message and changes it into GlParameterList format in order to submit it to the LIMGR using the DmAsGwAsterProRequestor class.
- **DmAsGwAsterProRequestor** -- This class is derived off of the DmAsGwAsterRequestor class, and used to submit a query to the SDSRV via the LIM libs(ASTER->ECS)).
- **DmAsGwAsterRequest** -- Requests that take place from ASTER to ECS.
- **DmAsGwAsterRequestor** -- DataManagmentGatewayAsterRequestor.
- **DmAsGwAsterRSTAgg** -- This class is derived from the DmGwRSTAggregate class to add build functions that were not provided in the base class.
- **DmAsGwAsterRSTNonAgg** -- This class is derived from the DmGwRSTNonAggregate class to add build functions that were not provided in the base class.

library DmAsGwCommon

/ecs/formal/DM/ASTGW/src/Common

The DmAsGwCommon library contains untility classes and term definitions used by the Aster Gateway. Primarily, classes exist that translate between Aster and ECS terms storing commonly requested translations in cache, convert between dollars and yen, handle Aster Server event logging, and handle IK connection procedures used to transmit and receive ODL trees via sockets.

Classes

- **DmAsGwAdEntry** -- This class is identical to the parent DmAsGwCacheEntry class except that the destructor does a delete instead of a RecursiveDelete.

- **DmAsGwAdServices** -- This class aids in the translation of "ECS" IOS Advertising Service results GlParameterLists to "ASTER" GlParameterLists.
- **DmAsGwCache** -- This class is designed to provide a convenient mechanism for storing and retrieving CacheEntries.
- **DmAsGwCacheEntry** -- The DmAsGwCacheEntry class is designed to be used in conjunction with the DmAsGwCache class.
- **DmAsGwErrorMsg** -- This class is designed to be used in conjunction with error logging in order to reduce the number of lines of error logging code.
- **DmAsGwEvent** -- This class performs the logging events in Gateway managed server.
- **DmAsGwTranslate** -- This class is designed to translate between ECS and ASTER using the data dictionary.
- **DmGwServerConnection** -- This is a wrapper class that will take care of all of the IK connection procedures for transmitting and receiving odl trees via sockets.

library DmAsGwEcsReqProc

/ecs/formal/DM/ASTGW/src/EcsReqProc

The DmAsGwEcsReqProc library contains classes for all the ECS to ASTER request processing. These classes are used to send Browse, Inventory, Product, and Product Update request messages from ECS to ASTER. When the ECS request is submitted to ASTER, the constraints need to be extracted from the GlParameterList and put into an ODLTree for ASTER. After the request is processed, the results are put back into a GlParameterList and sent to ECS. The aggregate and nonaggregate classes used to create the ODLTrees are also in the DmAsGwEcsReqProc library.

Classes

- **DmAsGwEcsBrowseRequest** -- These functions are responsible for processing an ECS to ASTER Browse request.
- **DmAsGwEcsInvRequest** -- This class is responsible for processing an inventory request.
- **DmAsGwEcsProductUpdRequest** -- This class is a polymorphic specialization of the EcsRequest class for an Ecs-to-Aster PRODUCT_UPDATE_REQUEST.
- **DmAsGwEcsProRequest** -- This class is responsible for processing an acquire (product) request.
- **DmAsGwEcsRequest** -- This class is a base class from which specialized versions are derived.
- **DmAsGwResultAggregate** -- Adds an aggregate subtree to a tree.
- **DmAsGwResultNonAggregate** -- This class is used to extract information from an ODL tree.
- **DmAsGwSearchAggregate** -- Also, any methods needed for preparing an Aggregate structure for submission to the Aster Server that are not included in that parent class are included here.

- **DmAsGwSearchNonAggregate** -- This class is publicly inherited from DmGwRSTNonAggregate and contains routines for Building parameters that are not accounted for in the parent class.

library DmDdMsg

/ecs/formal/DM/DDICT/src/ddmsg

This library defines messages to be passed to the Data Dictionary.

Classes

- **DmDdResultMsg**
- **DmDdSchemaMsg** -- Creates message object and sends it to the DDICT for schema requests for modifications to the Data Dictionary database..
- **DmDdSearchMsg** -- Creates message object and sends it to the DDICT for Data Dictionary search requests..
- **DmDdStatusMsg** -- Creates message object and sends it to the DDICT to get the status of a requests previously sent to the DDICT.
- **DmDdUrRequestMsg** -- Creates message object and sends it to the DDICT for Data Dictionary UR requests..
- **DmDdUrResultMsg** -- The DDICT creates and returns result message object in response to an UR requests.

library DmDdReqProc

/ecs/formal/DM/DDICT/src/ddreqproc

This library defines various manipulations for the data dictionary.

Classes

- **DmDdArgInfoTbl** -- This class is used to store/process data regarding whether or not the table has the fields collectionId, siteId, and attributeId.
- **DmDdGlHelper** -- This class contains methods that help in the processing of a GlParameterList.
- **DmDdGroupTbl** -- This class is used to store/process data for groups.
- **DmDdJoinTable** -- This class is used to manipulate joins.
- **DmDdJPairsTbl** -- This class is used to store/process data for joinable pairs table information.
- **DmDdLevel1Tbl** -- This class is used to store/process data for level 1 information.
- **DmDdLevel2Tbl** -- This class is used to store/process data for joinable pairs table information.
- **DmDdLevel2TblEntry** -- This class is used to store/process data for level 2 table information.

- **DmDdLevel3Tbl** -- This class is used to store/process data for joinable pairs table information.
- **DmDdLevel3TblEntry** -- This class is used to store/process data for joinable pairs table information.
- **DmDdMapper** -- This class performs the valid mapping for all collections in the Data Dictionary.
- **DmDdParser** -- This class is intended to be used in parsing the request submitted to the DDICT.
- **DmDdProcMsg** -- This class is used to process messages for the Data Dictionary.
- **DmDdTables** -- This class is used to store/process data from the tables.
- **DmDdTblCols** -- This class is used to store/process data for joinable pairs table information.
- **DmDdValidsMapping** -- This class contains the entry for each mapping.
- **DmDdValidsMappingTable** -- This class is used to store/process data for valid mapping.
- **DmDdXrefTbl** -- This class is used to store/process data for cross-reference information.
- **DmDdXrefTblEntry** -- This class is used to store/process data for joinable pairs table information.

library DmGwV0If

/ecs/formal/DM/GatewayCSCI/src/V0If

The DmGwV0If library contains classes for all the ECS to V0 request processing. These classes are used to send Browse, Inventory, and Product request messages from ECS to V0. When the ECS request is submitted to V0, the constraints need to be extracted from the GlParameterList and put into an ODLTree for V0. After the request is processed, the results are put back into a GlParameterList and sent to ECS. The aggregate and nonaggregate classes used to create the ODLTrees for the result messages are also in the DmGwV0If library.

Classes

- **DmGwEcsAddress** -- This class is responsible for storing address information.
- **DmGwEcsAffiliation** -- This class is responsible for storing affiliation information.
- **DmGwEcsBrowseRequest** -- This class is responsible for processing a V0 Browse request.
- **DmGwEcsInvRequest** -- This class is responsible for processing a V0 Inventory request.
- **DmGwEcsLineItem** -- This class is responsible for storing line item information.
- **DmGwEcsODLAggregate** -- This class is responsible for building a sub-tree in ODL tree.
- **DmGwEcsODLNonAggregate** -- This class is responsible for inserting attribute values into ODL tree.
- **DmGwEcsProductRequest** -- This class is responsible for processing a V0 Product request.
- **DmGwEcsResultAggregate** -- This class is responsible for extracting sub-tree from ODL tree.

- **DmGwEcsResultNonAggregate** -- These functions are responsible for extracting values from ODL tree.
- **DmGwEcsServRequest** -- This class is a base class from which specialized versions are derived.

library DmGwV0Util

/ecs/formal/DM/GatewayCSCI/src/V0Util

The DmGwV0Util library contains the implementation of functions used by the ECS to V0 Gateway for client/server interfaces.

library DmLmAsterIfFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmAsterIfFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- **DmLmInterfaceFactory** -- This class creates an external interface object of a certain type.

library DmLmDbi

/ecs/formal/DM/LIMGR/src/dbi

The DmLmDbi library contains utilities for interfacing and connecting with the DDICT database. Querys can either be sumbitted as a GlParameter or a string.

Classes

- **DmLmDbiBase** -- This class contains utilities for interfacing with the Data Dictionary.
- **DmLmDbiUtils** -- This class contains utilities for interfacing with the Data Dictionary.
- **DmLmIntQuery** -- This class provides the functionality to Submit a query in string format to the data dictionary, then retrieve the results after submitting.

library DmLmDbiSh (shared)

/ecs/formal/DM/LIMGR/src/dbi

The DmLmDbi library contains utilities for interfacing and connecting with the DDICT database. Querys can either be sumbitted as a GlParameter or a string.

Classes

- **DmLmDbiBase** -- This class contains utilities for interfacing with the Data Dictionary.
- **DmLmDbiUtils** -- This class contains utilities for interfacing with the Data Dictionary.
- **DmLmIntQuery** -- This class provides the functionality to Submit a query in string format to the data dictionary, then retrieve the results after submitting.

library DmLmExtIf

/ecs/formal/DM/LIMGR/src/extIf

The DmLmExtIf library contains classes that allow the LIMGR to interface with external information managers. Specifically, the information managers are the ASTER IMS, Science Data Server, Document Server, Gateway, and V0 Gateway.

Classes

- **DmLmAsterServIF** -- This class allows the LIMGR external interface library to connect to the Aster IMS.
- **DmLmDataServerIF** -- This class object is a specialization of the DmLmExtIFBase class.
- **DmLmExtIFBase** -- This class object is the base class from which specialized versions (e.g. DmLmV0ServIF) are derived.
- **DmLmGranuleUR** -- The DmLmGranuleUR class represents a granule which is the result from inventory search.
- **DmLmGranuleUrProvider** -- Definition of the class DmLmGranuleUrProvider.
- **DmLmInfoMgr** -- This class represents the server for DMS.
- **DmLmInfoMgrIF** -- quest object representing this request.
- **DmLmV0ServIF** -- This class allows the ECS-V0 gateway to interface with the V0 data server.

library DmLmGlobal

/ecs/formal/DM/LIMGR/src/global

The global CSC provides classes for DM subsystem to handle the global variables. This is the both client and server sides library.

Classes

- **DmLmCollection** -- This class is an entry for the collection table.
- **DmLmCollectionTable** -- The class contains a vector which is called collection table.
- **DmLmConfiguration** -- Through the object, configuration of the information manager can be inquired, such as its name, type, UR, PF configuration file and etc.

library DmLmGlobalSh (shared)

/ecs/formal/DM/LIMGR/src/global

The global CSC provides classes for DM subsystem to handle the global variables. This is the both client and server sides library.

Classes

- **DmLmCollection** -- This class is an entry for the collection table.

- **DmLmCollectionTable** -- The class contains a vector which is called collection table.
- **DmLmConfiguration** -- Through the object, configuration of the information manager can be inquired, such as its name, type, UR, PF configuration file and etc.

library DmLmInfoMgrFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmInfoMgrFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- **DmLmInterfaceFactory** -- This class creates an external interface object of a certain type.

library DmLmInfoMgrFactorySh (shared)

/ecs/formal/DM/LIMGR/src/factory

The DmLmInfoMgrFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- **DmLmInterfaceFactory** -- This class creates an external interface object of a certain type.

library DmLmMapper

/ecs/formal/DM/LIMGR/src/mapper

The **DmLmMapper** library contain a class used to map client attribute names to science data server attribute names and map ECS attribute names and values to VO attribute names and values.

Classes

- **DmLmMapper** -- The class provides methods to map attribute names and values from ECS to V0 or ASTER term, or vice versa.

library DmLmMapperSh (shared)

/ecs/formal/DM/LIMGR/src/mapper

The **DmLmMapper** library contain a class used to map client attribute names to science data server attribute names and map ECS attribute names and values to VO attribute names and values.

Classes

- **DmLmMapper** -- The class provides methods to map attribute names and values from ECS to V0 or ASTER term, or vice versa.

library DmLmMsg

/ecs/formal/DM/LIMGR/src/msg

The msg CSC provides classes for DM subsystem server and client. Client uses them to create the request objects sending to the server, and server uses them to create the result object replying to the client. This library is used both sides of client and server.

Classes

- **DmLmAcquireMsg** -- Creates message object and sends it to the SDSRV for acquire requests.
- **DmLmBrowseMsg** -- Creates message object and sends it to the SDSRV for browse requests.
- **DmLmExitMsg** -- Creates message object and sends it to the SDSRV for exit requests.
- **DmLmInspectMsg** -- Creates message object and sends it to the SDSRV for inspect requests.
- **DmLmNextBrowseMsg** -- Creates message object and sends it to the SDSRV for next browse requests.
- **DmLmResultMsg** -- The SDSRV creates and returns result message object inresponse to acquire, browse, exit, inspect, next browse, search, and status requests..
- **DmLmSearchMsg** -- Creates message object and sends it to the SDSRV for search requests.
- **DmLmStatusMsg** -- Creates message object and sends it to the SDSRV for status requests.

library DmLmReqProc

/ecs/formal/DM/LIMGR/src/reqproc

The reqproc CSC provides classes for DM subsystem server. It supplies the methods to handle the request. It is the main part of DM subsystem. This library is used in server side.

Classes

- **DmLmAcquireResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmAttributeCollectionsType**
- **DmLmBrowsePlan** -- This class provides methods to create request and integrate plans for the browse request.
- **DmLmBrowseResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmInspectPlan** -- This class provides methods to create request and integrate plans for the inspect request.
- **DmLmParsedReqInfo**
- **DmLmParsedReqInfoType** -- Information relevant to a parsed SEARCH/GUIDE request is stored in this class.

- **DmLmParser** -- The DmLmParser is a class for parsing and validating the request list.
- **DmLmPlanner** -- This class provides methods to create, request, and integrate plans for a search or product request.
- **DmLmProcessor** -- The public operations include to process the submition of the request to different target servers, create the result data store and integrate the results.
- **DmLmProcessPlan** -- This class is an entry of request processing.
- **DmLmProductPlan** -- This class provides methods to create request and integrate plans for the product request.
- **DmLmResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmSearchPlan** -- This class provides methods to create request and integrate plans for the search request.
- **DmLmSearchResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmSubmittedRequest** -- The class provides methods for request processing to submit a request to different types of target server.

library DmLmReqProcSh (shared)

/ecs/formal/DM/LIMGR/src/reqproc

The reqproc CSC provides classes for DM subsystem server. It supplies the methods to handle the request. It is the main part of DM subsystem. This library is used in server side.

Classes

- **DmLmAcquireResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmAttributeCollectionsType**
- **DmLmBrowsePlan** -- This class provides methods to create request and integrate plans for the browse request.
- **DmLmBrowseResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmInspectPlan** -- This class provides methods to create request and integrate plans for the inspect request.
- **DmLmParsedReqInfo**
- **DmLmParsedReqInfoType** -- Information relevant to a parsed SEARCH/GUIDE request is stored in this class.
- **DmLmParser** -- The DmLmParser is a class for parsing and validating the request list.
- **DmLmPlanner** -- This class provides methods to create, request, and integrate plans for a search or product request.

- **DmLmProcessor** -- The public operations include to process the submition of the request to different target servers, create the result data store and integrate the results.
- **DmLmProcessPlan** -- This class is an entry of request processing.
- **DmLmProductPlan** -- This class provides methods to create request and integrate plans for the product request.
- **DmLmResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmSearchPlan** -- This class provides methods to create request and integrate plans for the search request.
- **DmLmSearchResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmSubmittedRequest** -- The class provides methods for request processing to submit a request to different types of target server.

library DmLmReqProcStub

/ecs/formal/DM/LIMGR/src/reqprocStubs

Classes

- **DmLmAcquireResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmBrowseResultSet** -- This class provides methods to set or get information of results after a search request.
- **DmLmProcessPlan** -- This class is the stub class for testing the client.
- **DmLmResultSet** -- This class contains the final results obtained after processing.
- **DmLmSearchResultSet** -- This class provides methods to set or get information of results after a search request.

library DmLmV0IfFactory

/ecs/formal/DM/LIMGR/src/factory

The DmLmV0IfFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- **DmLmInterfaceFactory** -- This class creates an external interface object of a certain type.

library DmLmV0IfFactorySh (shared)

/ecs/formal/DM/LIMGR/src/factory

The DmLmV0IfFactory library contains classes that create the correct type of external interface used by the LIMGR to access external interfaces.

Classes

- **DmLmInterfaceFactory** -- This class creates an external interface object of a certain type.

library DpAt

/ecs/formal/PDPS/DPS/SSIT/src/LIB

This DpAt library holds the SSIT classes that are called by various SSIT tools and programs.

The DpAt library contains classes and methods that provide UNIX system call capabilities, GUI exception/error handling and Process Framework initialization/environmental setup.

Classes

- **DpAtMgrGuiException** -- This class manages GUI message popup windows.
- **DpAtMgrMessageDialog** -- This defines a Message Dialog box for the SSIT Manager and other SSIT GUIs.
- **DpAtMgrSystemCall** -- This class manages data for making system calls.
- **DpAtPfInit** -- This class's methods are used to initiate Process Framework in SSIT..
- **UIComponent** -- Base class for all C++/Motif UI components.

library DpAtDsrv

/ecs/formal/PDPS/DPS/SSIT/src/Dsrv

This library holds the object through which almost all SSIT calls to interface with the Data Server.

Classes

- **DpAtDsrv** -- This class's methods are the interface between the SSIT environment and the Data Server..

library DpAtMetadata

/ecs/formal/PDPS/DPS/SSIT/src/Metadata

This library holds the objects (**DpAtDatabase** and **DpAtOperationalMd**) through which most SSIT PDPS database updates are performed. It also conatins the object (**DpAtScienceMd**) which parses and validates the ODL that defines PGEs to PDPS.

The **DpAtDatabase** class is called by the ODL parsing programs to insert new PGE information and update existing PGEs in the PDPS database. **DpAtOperationalMd** provides checking and updating of the data from the Operational Metadata GUI. **DpAtScienceMd** provides ODL parsing and validation.

Classes

- **DpAtDatabase** -- This class's provides methods to update the PDPS database with PGE, ESDT and Production Rule operational and science metadata..

- **DpAtOperationalMd** -- This class's methods are callbacks for the PDPS/SSIT Database Update GUI, which updates the database with PGE operational metadata..
- **DpAtScienceMd** -- This class provides methods to read and validate all types of Science Metadata (ODL), PGE ODL, ESDT ODL and the various types of specific Production Rule ODL files.

library DpAtSSAP

/ecs/formal/PDPS/DPS/SSIT/src/SsapGui

This library contains the objects needed for the SSAP Editor application.

Classes

- **DpAtAccess** -- This class provides access control and access logs for the current user of the SSAP GUI.
- **DpAtBatchFile** -- This class provides access to SSAP commands in a file.
- **DpAtDirectoryInfo** -- This file contains the attributes and function prototypes of all operations of the DpAtDirectoryInfo class.
- **DpAtEditSSAPFileList** -- This class provides the functionality for adding & deleting files from the specified SSAP.
- **DpAtSSAPACFORM**
- **DpAtSSAPBatchModeFILE** -- This class defines the Batch File Window for the SSIT SSAP Editor GUI.
- **DpAtSSAPCreateFORM** -- This class defines the Create Window for the SSIT SSAP Editor Gui.
- **DpAtSSAPCreateWithFORM** -- This class defines the Create With window for the SSIT SSAP Editor GUI.
- **DpAtSSAPFrame** -- This class defines most of the SSIT SSAP Editor GUI.
- **DpAtSSAPManager** -- This class provides the primary access from SSIT to the Science Software Archive Packages stored at the Data Server.
- **DpAtSSAPMetadata** -- This class provides access from SSIT to the Science Software Archive Package metadata.
- **UIComponent**

library DpDeletionClient

/ecs/formal/PDPS/DPS/PRONG/src/DataMgmt/DeletionPF

The library containing the client side operations to issue requests to the deletion server.

This library contains two client usable APIs: one to issue a delete request on short and long delete times and another to delete a list of granules from the PDPS database. This first API is used by EcDpPrDM in the destage job, to issue deletes for interim granules. The second API is used by Production Request Editor, when deleting DPRs (and their associated granules).

Classes

- **Deletion_1_0**
- **DpDeletionProxy** -- Converts the data types into the required OODCE formats before calling the underlying OODCE client-side function and converts the return DCE data back to what's required.

library DpPrDbPf

/ecs/formal/PDPS/DPS/PRONG/src/Common

This library contains the PDPS database interface classes for use with the Process Framework access to the database.

Classes

- **DpPrDbColVal** -- This class is used to create a column-value structure so that the database interface class can use to map the object attribute to appropriate column in the table.
- **DpPrDbColValList** -- This class contains a list of DpPrDbColVal objects.
- **DpPrDbConnectRecord**
- **DpPrDbIF** -- Must undefine external definition of Status before defining it below.
- **DpPrDbMaster** -- This base class monitors all the connections to the database.

Templates

- **DpPrDbInterface** -- This class provides neccessary methods to manipulate database table from C++ application program.

library DpPrDM

/ecs/formal/PDPS/DPS/PRONG/src/DataMgmt

This library supplies utility operations to manage data within DPS.

These operations are used for the two threads within the administrative job, EcDpPrDM: those of staging and destaging data, and managing their disc resources via calls to resource manager. There are no static operations, so an object must be instantiated to take advantage of the public operations available.

Classes

- **DpPrDataGransGroup** -- This class defines attributes and operations for a group of data granules based on groupId and DataType name.
- **DpPrDataManager** -- This class defines attributes and operations for initializing and managing data granules required by a PGE during its execution.
- **DpPrDMPfClient** -- This class invokes stage and destage jobs in the AUTOSYS job box.
- **DpPrFile** -- Since this class is a persistent class, it defines attributes and operations for manipulating the instances of objects as the entries in the Sybase DataBase.

- **DpPrGranuleLocator** -- This class provides an interface to the DpPrGranuleLocation table and some convenience methods for determining whether or not data is local (in the DAAC and visible), nearby (in the DAAC but not visible), or need to be staged from the DSS.

library DpPrDMStub

/ecs/formal/PDPS/DPS/PRONG/src/DataMgmt

This library contains stubs for the public operations called for data management within DPS.

Provided to allow easier unit testing of applications which use the real library, this library makes no calls and performs no real operations, merely passes back a status to the calling function.

Classes

- **DpPrDataGransGroup** -- This class defines attributes and operations for a group of data granules based on groupId and DataType name.
- **DpPrDataManager** -- This class defines attributes and operations for initializing and managing data granules required by a PGE during its execution.
- **DpPrDMPfClient** -- This class invokes stage and destage jobs in the AUTOSYS job box.
- **DpPrFile** -- Since this class is a persistent class, it defines attributes and operations for manipulating the instances of objects as the entries in the Sybase DataBase.
- **DpPrGranuleLocator** -- This class provides an interface to the DpPrGranuleLocation table and some convenience methods for determining whether or not data is local (in the DAAC and visible), nearby (in the DAAC but not visible), or need to be staged from the DSS.

library DpPrDssIF

/ecs/formal/PDPS/DPS/PRONG/src/DssIF

This library contains the helper object for PDPS to interact with science data server.

Clients use this object, rather than making a direct call, so that the interface can be encapsulated within a single class for maintainability. This library includes calls to APIs from both DSS and IOS.

Classes

- **DpPrDSSInterface** -- This class provides a central (NON-PF-CLIENT) class for all interactions with either DSS or the DSS simulator.
- **DpPrRpcID** -- This object encapsulates the EcUtRpcID for PDPS.

library DpPrEM

/ecs/formal/PDPS/DPS/PRONG/src/ExecMgmt

This library supplies utility operations to manage PGE execution within PRONG.

This library is mostly used by DpPrEM.cxx, which creates the program EcDpPrEM. Using this library, through the instation of **DpPrExecutionManager** and using its public API, allocation, deallocation, pre and post processing can be accomplished when running a PGE.

Additionally, using the **DpPrPge** class, the current state of a particular dpr, whether it has finished allocating resources or whether it has post processed the pge's outputs etc., can be obtained.

Also, several of **DpPrUtil** class's methods are static methods. These methods can be used to execute a system command, such as to obtain unix file information.

Classes

- **DpPrExecutable** -- This class is used to maintain the state of the science software components which are crucial to support the proper runtime operation of a PGE.
- **DpPrExecutionManager** -- This DpPrExecutionManager class is the interface class for other classes which require execution services.
- **DpPrLock** -- This class provides general purpose semiphore for critical sections of code.
- **DpPrPcf** -- This class is used to maintain the state of the Process Control File (PCF), which provides critical runtime information to the PGE, while the science software is executing.
- **DpPrPcfGran** -- DpPrPcfGran represents a granule (inputs & outputs).
- **DpPrPge** -- This class is used to maintain the state of the completed PGE and as such provides for the insertion and deletion of the science software for the local machine and controls execution of the PGE on that platform.
- **DpPrUtil** -- This class provides utility methods used by EM.
- **PlCompletion** -- This class is used to maintain the termination results of a DPR run.

library DpPrEMStub

/ecs/formal/PDPS/DPS/PRONG/src/ExecMgmt

This library supplies stubbed operations to manage PGE execution within PRONG. It is identical to DpPrEM, except all interfaces to science data server and Advertising server have been stubbed out.

Classes

- **DpPrExecutable** -- This class is used to maintain the state of the science software components which are crucial to support the proper runtime operation of a PGE.
- **DpPrExecutionManager** -- This DpPrExecutionManager class is the interface class for other classes which require execution services.
- **DpPrLock** -- This class provides general purpose semiphore for critical sections of code.
- **DpPrPcf** -- This class is used to maintain the state of the Process Control File (PCF), which provides critical runtime information to the PGE, while the science software is executing.
- **DpPrPcfGran** -- DpPrPcfGran represents a granule (inputs & outputs).
- **DpPrPge** -- This class is used to maintain the state of the completed PGE and as such provides for the insertion and deletion of the science software for the local machine and controls execution of the PGE on that platform.

- **DpPrUtil** -- This class provides utility methods used by EM.
- **PlCompletion** -- This class is used to maintain the termination results of a DPR run.

library DpPrException

/ecs/formal/PDPS/DPS/PRONG/src/Exception

This library provides APIs to catch and handle signals.

Exception is a misleading name. This library does not catch C++ exceptions. It catches unix signals (mainly SIGHUP) and programs which use this library can perform any cleanup before exiting.

Classes

- **DpPrInterrupt** -- This class manages PRONG administrative job's process interrupt state. It manipulates the process's interrupt signal handling characteristics to provide a transparent mechanism for a running process to perform a graceful shutdown when requested.

library DpPrJM

/ecs/formal/PDPS/DPS/PRONG/src/JobMgmt

The Job Management library provides service routines for the Job Management Server to use to manage the flow of jobs in all AutoSys instances for a mode. These services include creating jobs, canceling jobs, releasing queued jobs, getting the status of jobs and controlling job execution through autosys. These routines are not called directly by clients, instead, a proxy library is provided to clients that have interfaces similar to the ones in routines in this library.

Classes

- **DpPrAutosysMap** -- This class assigns and retrieves the maps between resource strings and the IDs of AutoSyses managing jobs running on them.
- **DpPrAutosysMapList** -- This persistent class stores and provides access methods to the list of AutoSys ID <-> string name mappings.
- **DpPrCotsManager** -- This class provides an interface to the AutoSys COTS product to add, delete, and modify jobs and machines; control job processing; retrieve job definition and processing status; generate reports; and send alarms.
- **DpPrCreateQObj** -- This class contains information necessary for constructing AutoSys jobs.
- **DpPrCreationQueue** -- This class manages queues of CreateDprJob parameters waiting to be used to create jobs in AutoSys.
- **DpPrCreationQueueByAutosys** -- This is a priority queue (sorted vector) of CreateDprJob parameter lists.
- **DpPrDbTableIf** -- This class defines the interface to Sybase for the DpPrDbInterface class.
- **DpPrJIL** -- AutoSys JIL interface.

- **DpPrScheduler** -- This class provides PLANG Planning Workbench and Subscription Manager components a programmatic interface to the PRONG.

library DpPrQaMonitor

/ecs/formal/PDPS/DPS/PRONG/src/QaMonitor

This library holds the functions and attributes that all the QA Montior GUI to perform its various functions.

The actual GUI code is in a separate section, the classes within this library provide the function to the QA Montior GUI.

Classes

- **DpPrQaDataGranule** -- It defines attributes and operations of data parameters returned from DataServer when operator tries to query data for a particular data type during a period of starting and ending day.
- **DpPrQAGranuleQaFlags** -- These methods and attributes support display and update of granule parameters.
- **DpPrQaMonitor** -- The DpPrQaMonitor is what the Quality Assurance (QA) position uses to query all data granules for a particular data type within a date interval, retrieve data granules from Data Server to local disk, and to view data products created with QA metadata

library DpPrRM

/ecs/formal/PDPS/DPS/PRONG/src/RsrcMgmt

This library supplies utility operations to manage computer resources for PDPS.

There is only ever one instance of a Resource Manager active at any one time. A resource lock prevents other operations from creating multiple Resource Manager objects.

Classes

- **DpPrComputer** -- This class is used to represent the set of computer hardware that is being used for science software processing within the Processing System.
- **DpPrCpuRamAllocation** -- This class is used to maintain the individual records of CPU and RAM usage, which are used to maintain the integrity of processor and memory allocations and deallocations.
- **DpPrDiskAllocation** -- This class is used to maintain the individual records of disk storage usage which are used to maintain the integrity of storage allocations and deallocations.
- **DpPrDiskPartition** -- This class is used to represent the set of disk storage devices that are being used to contain the files needed to run a collective set of software known as a PGE.
- **DpPrResource** -- This base class is used to capture the similar features of the derived classes, and to provide for future expansion.
- **DpPrResourceLock** -- This class is used to implement locking of the Resource Management database tables on a per-job basis.

- **DpPrResourceManager** -- This interface class provides an abstract set of operations for effectively managing the collection of processing and storage resources.
- **DpPrString** -- Abstract representation of one or more actual machines.

library DpPrTools

/ecs/formal/PDPS/DPS/PRONG/src/Tools

This library contains global utility functions.

This library contains global utility functions to get directory listing, space available on a given device, ability to set the application log file name etc.,

library DsAdSh (shared)

/ecs/formal/DSS/sdsrv/src/ad

This library provides an administrative/M&O interface to science data server requests. It also provides a collection mechanism for retrieving and manipulating multiple descriptors. It is mainly used by EcDsSdSrvGui for installing ESDTs and monitoring the Science Data Server requests.

Classes

- **DsAdDataTypeCollector** -- This class provides a collection mechanism for retrieving and manipulating multiple Earth Science Data Type descriptors.
- **DsAdSciRequest** -- Provides an interface to access Science Data Server Request information like request ids, priority, domain and state request information as well as command service, command category and number of commands.
- **DsAdSciRequestInterface** -- This class provides an administrative/M&O interface to science data server requests.

library DsBtSh (shared)

/ecs/formal/DSS/sdsrv/src/bt

This library implements a thread which notifies the subscription server of events. Notification requests are inserted in the queue by the processing threads in the data server and are removed and processed by this thread in the background. This processing is done in a separate thread so that if the subscription server is not responding, the processing of the request can proceed, and the sending of the event can be retried ad infinimum until the subscription server comes back. In addition, the queue is mirrored in the databawse, so that if the dataserver must be restared before the events are sent, the queue will not be lost.

Classes

- **DsBtSbsrvNotifier** -- This class (actually a collection of static functions and variables) implements a thread which notifies the subscription server of events.

library DsCl

/ecs/formal/DSS/sdsrv/src/cl

This library contains classes which make up the Science Data Server's (SDSRV) public interface. These classes are used by clients to connect,build commands, and submit requests to the SDSRV.

Classes

- **DsClAcquireCommand** -- Provides an encapsulated mechanism for transporting ESDT acquire command parameters through the SDSRV client interface.
- **DsClCommand** -- Provides a client view of the shared command implementation class, which is transferred to the server upon submission.
- **DsClDescriptor** -- Client side class that provides an interface to the descriptor.
- **DsClESDTAddRequest** -- This class is used to add an ESDT Reference to the ESDTReferenceCollector as well as adding an ESDT to the working collection.
- **DsClESDTReference** -- This object is a reference to an ESDT that is within a Data Server's holdings. This object provides services that are homogeneous for all ESDTs..
- **DsClESDTReferenceCollector** -- This class provides the primary interaction mechanism for client software.
- **DsClESDTReferenceCollectorVector** -- This object is a vector that contains a set of ESDT Reference Collectors.
- **DsClESDTReferenceVector** -- ESDT type-level info + ptrs to ESDTs.
- **DsClESDTRemoveRequest** -- This class is used to remove an ESDT Reference from the working collection.
- **DsClGenConnector** -- This class encapsulates communications with a data server.
- **DsClGenRequestInt** -- Represents the generic portion of the client side's interface to request objects..
- **DsClGenSuspendor** -- Provides specialized collector types requiring state management.
- **DsClInsertCommand** -- Provides an encapsulated mechanism for transporting ESDT insert command parameters through the SDSRV client interface.
- **DsClQuery** -- Provides a convenient way for clients to make search requests.
- **DsClRequest** -- It is composed of a sequence of DsClCommand objects (created by the client), which are the basic execution units.
- **DsClRestartNotifier** -- This class is used by the client to notify the Science Data Server of its start up temperature.
- **DsClSubsetCommand** -- DsClSubsetCommand is a helper class inherited from DsClCommand.
- **DsClTypeInfo** -- Container class for product-type-level metadata.

library DsClSh (shared)

/ecs/formal/DSS/sdsrv/src/cl

This library contains classes which make up the Science Data Server's (SDSRV) public interface. These classes are used by clients to connect,build commands, and submit requests to the SDSRV.

Classes

- **DsClAcquireCommand** -- Provides an encapsulated mechanism for transporting ESDT acquire command parameters through the SDSRV client interface.
- **DsClCommand** -- Provides a client view of the shared command implementation class, which is transferred to the server upon submission.
- **DsClDescriptor** -- Client side class that provides an interface to the descriptor.
- **DsClESDTAddRequest** -- This class is used to add an ESDT Reference to the ESDTReferenceCollector as well as adding an ESDT to the working collection.
- **DsClESDTReference** -- This object is a reference to an ESDT that is within a Data Server's holdings. This object provides services that are homogeneous for all ESDTs..
- **DsClESDTReferenceCollector** -- This class provides the primary interaction mechanism for client software.
- **DsClESDTReferenceCollectorVector** -- This object is a vector that contains a set of ESDT Reference Collectors.
- **DsClESDTReferenceVector** -- ESDT type-level info + ptrs to ESDTs.
- **DsClESDTRemoveRequest** -- This class is used to remove an ESDT Reference from the working collection.
- **DsClGenConnector** -- This class encapsulates communications with a data server.
- **DsClGenRequestInt** -- Represents the generic portion of the client side's interface to request objects..
- **DsClGenSuspendor** -- Provides specialized collector types requiring state management.
- **DsClInsertCommand** -- Provides an encapsulated mechanism for transporting ESDT insert command parameters through the SDSRV client interface.
- **DsClQuery** -- Provides a convenient way for clients to make search requests.
- **DsClRequest** -- It is composed of a sequence of DsClCommand objects (created by the client), which are the basic execution units.
- **DsClRestartNotifier** -- This class is used by the client to notify the Science Data Server of its start up temperature.
- **DsClSubsetCommand** -- DsClSubsetCommand is a helper class inherited from DsClCommand.
- **DsClTypeInfo** -- Container class for product-type-level metadata.

library DsCn

/ecs/formal/DSS/sdsrv/src/cn

This library was used to generate files which install ESDT configuration information for each ESDT previously. Presently the database is used instead to store the configurations. Therefore the library will not be used by SDSRV, but it is still used by STMGT.

Classes

- **DsCnConfiguration** -- Used to access a server specific configuration file.

library DsCnSh (shared)

/ecs/formal/DSS/sdsrv/src/cn

This library was used to generate files which install ESDT configuration information for each ESDT previously. Presently the database is used instead to store the configurations. Therefore the library will not be used by SDSRV, but it is still used by STMGT.

Classes

- **DsCnConfiguration** -- Used to access a server specific configuration file.

library DsCsSh (shared)

/ecs/formal/DSS/sdsrv/src/cs

This library is used by the HDFEOS server which supplies subsetting services to DLL or other clients. Access to the subsetting is via Cs client calls to the server pool managed by the Science Data Server. This library contains the low level HDFEOS server code responsible for Conformant and NonConformant subsetting, metadata update for subsetted granule, and HDF-EOS reformatting. The DsCs library is dependent on Hc, Hr and l7_dcs libraries.

Classes

- **DsCsAttribute** -- This class holds all the attribute information of a HDFEOS object.
- **DsCsConformant** -- This class implements operations for the Conformant interface.
- **DsCsConformantImp --** This class implements operations for the Conformant file interface.
- **DsCsCSDT** -- This is an abstract base class of the classes DsCsConformant and DsCsNonConformant.
- **DsCsCSDTImp** -- This class is an abstract superclass of the DsCsHDFEOSImp and DsCsNonHDFEOSImp classes.
- **DsCsDimMap** -- This class fetches the mappings from the HDFEOS input file and can define them in the output HDFEOS file.
- **DsCsDimMgr** -- This is an abstract class which is the superclass of the classes DsCsSwathMgr and DsCsGridMgr.
- **DsCsDimSum** -- This is a summary of all dimension information.

- **DsCsField** -- This class stores all information related to a HDFEOS field.
- **DsCsGridMgr** -- This class is responsible for creating grids resulting from subsetting, including creating new grid names, dimension names, field names and dimension map names etc..
- **DsCsHDFEOSImp** -- This class provides a placeholder for the grid, swath and point data objects in HDFEOS files.
- **DsCsHdfEosSDS**
- **DsCsHDFEOSUtil** -- This class is for utilities common to HDFEOS users.
- **DsCsIdxMap** -- This class fetches the indexed mappings from HDFEOS input file and defines them in the output HDFEOS file.
- **DsCsMask** -- All masks which are available for inter-DAAC compression are stored as members of this class.
- **DsCsMetadataUpdate** -- This class implements updating the inventory metatdata in the HDFEOS output file after subsetting calls made to the HDFEOS server, as well as copying necessary attributes (e.g. global attributes) and their values to the subsetted output file.
- **DsCsMgr** -- This is an abstract class which is the superclass of the classes DsCsDimMgr and DsCsPointMgr.
- **DsCsNonConformant** -- This interface is for products which do not use the naming conventions in the subsetting white papers.
- **DsCsNonConformantImp** -- This class implements operations for the NonConformant file subsetting interface.
- **DsCsNonHDFEOSImp** -- This class provides a placeholder for non-HEDEOS data objects in the file.
- **DsCsOutputDimSDS** -- This class provides the required interfaces with Grid and Swath objects in HDFEOS as well as SDS in HDF output files.
- **DsCsOutputFile** -- This class is used by the DsCsHDFEOSImp class and provides operations that interface with staging disk classes for storage management.
- **DsCsOutputGrid** -- This class provides all required interfaces (defining, creating, writing etc.) with grid data objects in HDFEOS output files.
- **DsCsOutputPoint** -- This class provides all required interfaces (defining, creating, writing etc.) with point data objects in HDFEOS output files.
- **DsCsOutputSDS**
- **DsCsOutputSwath** -- This class provides all required interfaces (defining, creating, writing etc.) with swath data objects in HDFEOS output files.
- **DsCsPointMgr** -- This class is responsible for creating the output points based on the constraint parameters.
- **DsCsProjection** -- This class is used to extract, store and define grid projection information like projection type, zone code, spheroid type, projection parameters, pixel location and origin of data.

- **DsCsPtLevel** -- DsCsPtLevel class is responsible for storing all the level information for a level in the output point.
- **DsCsSrcDimSDS** -- This is a superclass with subclasses DsCsSrcGrid and DsCsSrcSwath.
- **DsCsSrcGrid** -- This class provides all required interfaces (inquiring, reading etc.) with grid data objects in HDFEOS source file.
- **DsCsSrcPoint** -- This class provides all required interfaces (inquiring, reading, extracting etc.) with grid data objects in HDFEOS source file.
- **DsCsSrcSDS** -- This is a superclass with subclasses DsCsSrcDimSDS and DsCsSrcPoint.
- **DsCsSrcSwath** -- This class provides all required interfaces (inquiry, reading, extracting etc.) with swath data objects in HDFEOS source files.
- **DsCsSwathMgr** -- This class is responsible for creating swaths resulting from subsetting, including creating new swath names, dimension names, field names and dimension map names etc..
- **ESDTAnyProduct** -- This is the public interface used by the Client to provide subsetting functionality.

library DsDb

/ecs/formal/DSS/sdsrv/src/db

This library encapsulates the Sybase Open Client calls from the rest of the SDSRV. All SDSRV database access is made through this this library (both dynamic SQL and Stored Procedure calls). As well as **DsDbInterface**, this library also contains code for the DsDbAttributeToTable, **DsDbAttributeTableXref**, **DsDbAttributeToTableVector**, DsDbProductDbXref, DsDbGranuleToDbVector, and ResultDesc.

Classes

- **DsDbAttributeTableXref** -- Attribute table entry class.
- **DsDbAttributeToTableVector** -- Attribute to table hash table class.
- **DsDbInterface** -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.

library DsDbSh (shared)

/ecs/formal/DSS/sdsrv/src/db

This library encapsulates the Sybase Open Client calls from the rest of the SDSRV. All SDSRV database access is made through this this library (both dynamic SQL and Stored Procedure calls).

Classes

- **DsDbAttributeTableXref** -- Attribute table entry class.
- **DsDbAttributeToTableVector** -- Attribute to table hash table class.

- **DsDbInterface** -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.

library DsDcSh (shared)

/ecs/formal/DSS/sdsrv/src/l7_dcs

This library is used by subsetting code which make reformatting calls to convert HDF-EOS file format to HDF file format for subsetted HDF files. At present, it is only for Landsat7 data conversion.

library DsDdBSh (shared)

/ecs/formal/DSS/distribution/src/base

This library contains base objects used in both the client and server components of DDIST.

Classes

- **DsDdDbConnectionPool** -- DDIST base class for the database connection pool.
- **DsDdDbInterface** -- Database Interface class.
- **DsDdDistFileB** -- Holds the client side file-level information for the distribution request.
- **DsDdDistListB** -- Implements the client side of the distribution request, holding the list of granules for the request.
- **DsDdDistRequestB** -- DDIST base class for the client-server requests; holds information relating to the request.
- **DsDdGranuleB** -- Holds the client side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.
- **DsDdRequestListB** -- The DDIST base class for the client-server request list.

library DsDdCSh (shared)

/ecs/formal/DSS/distribution/src/client

This is the client side library.

Classes

- **DsDdDistFileC** -- Holds the client side file-level information for the distribution request.
- **DsDdDistListC** -- Implements the client side of the distribution request, holding the list of granules for the request.
- **DsDdDistRequestC** -- DDIST client side of request client-server interface; holds information relating to the request.
- **DsDdEventLogC** -- Implements the client side of the Event Logging error/messages in to the data distribution database.
- **DsDdEventLogListC** -- Implements the client side of the Event Log's client list..

- **DsDdGranuleC** -- Holds the client side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.
- **DsDdRequestListC** -- DDIST class for the client request list.
- **DsDdRequestMgrBaseC** -- Client-side request list for DDIST client-server interface.
- **DsDdRequestMgrC** -- Client-side request list subclass for DDIST client-server interface.
- **DsDdRequestMgrCFhExecutor**
- **DsDdRequestMgrCProxy** -- This class provides a DCE-independent interface to the widget on the client-side.
- **DsDdTapeIdC** -- Implements the client side of the Tape ID's associated with a distribution request..
- **DsDdTapeIdListC** -- Implements the client side of the Tape ID's client list associtated with a distribution request..
- **DsDdTapeRequestListC** -- DDIST class for the client 4mm or 8mm request list.
- **DsDdTestSuite** -- The DsDdTestSuite is a collection of test methods collected into a class for ease of maintenance and testing.

library DsDdISh (shared)

/ecs/formal/DSS/distribution/src/idl

Classes

- **DsDdRequestMgrIDL_2_0**
- **DsDdRequestMgrIDL_2_0_ABS**
- **DsDdRequestMgrIDL_2_0_Mgr**

library DsDdSSh (shared)

/ecs/formal/DSS/distribution/src/server/main

Classes

- **DsDdActiveQueue** -- This class is the class for the queue of active data distribution requests.
- **DsDdAuxState**
- **DsDdBaseQueue** -- This class is the base class for queues.
- **DsDdCBCache** -- This class stores the capacity and block size for each media type.
- **DsDdConfiguration** -- The DsDdConfiguration class is a singleton class which manages configuration information for the DDIST subsystem.
- **DsDdDCERequestMgrConcrete** -- This class provides (server-side) definitions for the distributed functions for a widget.
- **DsDdDistFileS** -- Holds the client side file-level information for the distribution request.

- **DsDdDistListS** -- Implements the server side of the distribution request, holding the list of granules for the request and estimation information.
- **DsDdDistRequestS** -- DDIST server side of request client-server interface; holds information relating to the request.
- **DsDdDistStatus** -- The DsDdDistStatus class defines the status information associated with a distribution request.
- **DsDdDoneQueue** -- This class is the class for the queue of shipped data distribution requests.
- **DsDdGranuleS** -- Holds the server-side granule-level information and the list of files which comprise the granule; files can be from archives or working areas under STMGT control.
- **DsDdHoldQueue** -- This class is the class for the queue of holding data distribution requests.
- **DsDdLog** -- Dont forget to inline.
- **DsDdMedia** -- Base class for defining media-specific processing associated with a particular media type.
- **DsDdMediaBuilder** -- Handles loading the shared object executable file for media-specific processing and creating an object instance of the specific class.
- **DsDdMediaDist** -- DDIST packaging class which holds a list of granules which will be distributed on the media.
- **DsDdMediaFactory** -- The DsDdMediaFactory class is a singleton class which handles the creation of DsDdMedia-based classes by loading and managing entry points to DLLs which hold each of the DsDdMedia-based class code, structure, constructors, destructors, methods, etc.
- **DsDdPfConfigFile** -- The DsDdPfConfigFile class is a singleton class.
- **DsDdPriorityQueue** -- This class is the class for the queue of pending data distribution requests.
- **DsDdPriorityThread**
- **DsDdReadyToShipQueue** -- This class is the class for the queue of data distribution requests which are ready to ship.
- **DsDdRequestListS** -- DDIST class for the server request list.
- **DsDdRequestMgrReal** -- This class represents the server-side DCE-independent widget.
- **DsDdRequestMgrServer**
- **DsDdRequestStatus** -- Class for defining the status of a request from the view of DDist down.
- **DsDdScheduler** -- This class is responsible for multi-threading requests.
- **DsDdSetLogFname**
- **DsDdTestSuiteS** -- The DsDdTestSuite is a collection of test methods collected into a class for ease of maintenance and testing.

library DsDe1Sh (shared)

/ecs/formal/DSS/sdsrv/src/de1

This internal library contains the method to do validation, installing ESDT, sending metadata to IOS,DMS and SBSRV and generate MCF to ingest.

Classes

- **DsDeCustomizer** -- Base class that provides the ability to customize features of an ESDT.
- **DsDeCustomizerVector** -- This is a container class for customizers for a given datatype.
- **DsDeDataDictController** -- This class encapsulates SDSRV's interaction with DDICT. Policy decisions are placed here to unburden the Descriptor classes..
- **DsDeDDCustomizer** -- This class provides the ability to customize the data dictionary information for an ESDT.
- **DsDeDescriptor** -- This is the server side class that provides an interface to the descriptor.
- **DsDeDescriptorBuilder** -- This class reads in the ASCII file that defines the descriptor for a particular type.
- **DsDeDescriptorImp** -- This class contains all of the attributes and associated implementation of the descriptor class.
- **DsDeDescriptorSet** -- DsDeDescriptorSet is the keeper of all descriptors.
- **DsDeEventCustomizer** -- This class provides the ability to configure the an event recognized by the ESDT.
- **DsDeEventTable** -- This class holds the cross reference of event names to event IDs.
- **DsDeExpressionRule** -- This file only contains stuff used by both kinds of expression rules so far it is simply a function that validates operators.
- **DsDeIOSController** -- This class encapsulates SDSRV's interaction with ADVSRV. Policy decisions are placed here to unburden the Descriptor classes..
- **DsDeMetadataCustomizer** -- This class provides the ability to customize metadata attributes for an ESDT.
- **DsDeMetadataCustomizerVector** -- This is a container class for the set of metadata customizers for a given datatype.
- **DsDeRealRuleFactory** -- This class is a specialization of the RuleFactory class.
- **DsDeRule** -- This is an abstract base class for all validation rules.
- **DsDeRuleFactory** -- This class provides the interface by which all rules are created.
- **DsDeServiceCustomizer** -- Provides the ability to customize the services that an ESDT recognizes.
- **DsDeServiceCustomizerVector** -- This is a container class for the set of service customizers for a given datatype.
- **DsDeStructCustomizer** -- This class provides the ability to customize the CSDT type and format that an ESDT is implemented with.

- **DsDeStructCustomizerVector** -- This is a container class for the set of struct customizers for a given datatype.
- **DsDeSymbolicExpressionRule** -- This class is a specialization of the SymbolicRule class.
- **DsDeSymbolicMatchRule** -- This class is a specialization of the SymbolicRule class.
- **DsDeSymbolicRangeRule** -- This class is a specialization of the SymbolicRule class.
- **DsDeSymbolicRule** -- This is an abstract base class that provides the interface for creating concrete symbolic rules.
- **DsDeSymbolicRuleFactory** -- This class is used to create symbolic rules of the correct type.

Templates

- **DsDeRealExpressionRule** -- This template class is a specialization of the RealRule class.
- **DsDeRealMatchRule** -- This template class is a specialization of the RealRule class.
- **DsDeRealRangeRule** -- This template class is a specialization of the RealRule class.

library DsDe2Sh (shared)

/ecs/formal/DSS/sdsrv/src/de2

This internal library contains the method to do validation, installing ESDT, sending metadata to IOS,DMS and SBSRV and generate MCF to ingest.

Classes

- **DsDeAttNode** -- Holds a single metadata attribute and provides services with respect to that attribute.
- **DsDeConvert** -- Used to do various types of conversions that are needed by the descriptor.
- **DsDeDDAttribute** -- Holds the dictionary information for DID 311 attributes.
- **DsDeDDContent** -- Holds the dictionary information for container objects.
- **DsDeDictionary** -- This class is used to hold the DID311 dictionary information.
- **DsDeESDTCreator** -- This class is used to add a new ESDT to the system.
- **DsDeGroupNode** -- Provides services with respect to the set of attributes.
- **DsDeMCFBuilder** -- Writes the MCF to a stream and also is capable of filtering out sections of the MCF.
- **DsDeMCFtoODL** -- Generates the MCF file in ODL.
- **DsDeNode** -- Holds any kind of metadata attribute part and is used to make arbitrary nested metadata constructs.
- **DsDePSGroup** -- Holds product specific metadata attributes.
- **DsDePSNode** -- Provides services for a single product specific metadata attribute.
- **DsDeRealRule** -- Provides the interface to creating concrete rules.

library DsGe

/ecs/formal/DSS/sdsrv/src/ge

This library provides the public interface for access to all ESDT services and ESDT specifications. It determines ESDT configuration information. It also performs runtime loading of ESDT libraries for a data type.

Classes

- **DsGeDynamicLibrary** -- Class that implements the dynamic library loading.
- **DsGeECSDataProduct** -- Inherit from the public ESDT class, which provides basic ESDT functionality.
- **DsGeESDT** -- Inherit from the public class - provides basic ESDT functionality.
- **DsGeESDTConfiguration** -- DsGetESDTConfiguration.
- **DsGeESDTConfiguredType** -- This struct is used to hold the information for a ESDT type.
- **DsGeESDTDynamicLibrary** -- This class performs runtime loading of ESDT libraries for a data type.
- **DsGeESDTServiceProvider** -- This abstract base class provides all of the truly public methods for ESDTs and ESDT Wrappers.
- **DsGeESDTWrapper** -- This class encapsulates the ESDT functionality and provides the public interface for access to all ESDT services.
- **DsGeMetadataUpdate** -- This class models the update of metadata, specifically for update during a subsetting operation on a ESDT object.
- **DsGeOID** -- This class provides a mechanism for uniquely identifying a data granule within a DataServer.
- **DsGeScienceData** -- Inherit from the public ESDT class, which provides basic ESDT functionality.

library DsGeSh (shared)

/ecs/formal/DSS/sdsrv/src/ge

This library provides the public interface for access to all ESDT services and ESDT specifications. It determines ESDT configuration information. It also performs runtime loading of ESDT libraries for a data type.

Classes

- **DsGeDynamicLibrary** -- Class that implements the dynamic library loading.
- **DsGeECSDataProduct** -- Inherit from the public ESDT class, which provides basic ESDT functionality.
- **DsGeESDT** -- Inherit from the public class - provides basic ESDT functionality.
- **DsGeESDTConfiguration** -- DsGetESDTConfiguration.

- **DsGeESDTConfiguredType** -- This struct is used to hold the information for a ESDT type.
- **DsGeESDTDynamicLibrary** -- This class performs runtime loading of ESDT libraries for a data type.
- **DsGeESDTServiceProvider** -- This abstract base class provides all of the truly public methods for ESDTs and ESDT Wrappers.
- **DsGeESDTWrapper** -- This class encapsulates the ESDT functionality and provides the public interface for access to all ESDT services.
- **DsGeMetadataUpdate** -- This class models the update of metadata, specifically for update during a subsetting operation on a ESDT object.
- **DsGeOID** -- This class provides a mechanism for uniquely identifying a data granule within a DataServer.
- **DsGeScienceData** -- Inherit from the public ESDT class, which provides basic ESDT functionality.

library DsHcSh (shared)

/ecs/formal/DSS/sdsrv/src/hc

This library provides a DCE-independent client interfaces for both Conformant and NonConformant subsetting. It also provide a client class dervived from **EcPfClient** that can use Process Frame Work.

Classes

- **DsHcClient** -- This is the definition of HDF client that can use Process Frame Work.
- **DsHcConformant** -- Provides a DCE independent interface for Conformant Subsetting.
- **DsHcNonConformant** -- Provides a DCE independent interface for the NonConformant Subsetting.

library DsHrSh (shared)

/ecs/formal/DSS/sdsrv/src/hr

This library encapsulates all DCE exposure for the distributed objects and functions for both Conformant and NonConformant subsetting on the server side for HdfEos server. It also provides a client Proxy interface to the server's distributed objecs. It contains the mechanisms for asynchronization and rebinding.

Classes

- **DsHrAsynchRequest** -- It must be specialized to add the desired server side functionality.
- **DsHrConformantIDL_1_0**
- **DsHrConformantIDL_1_0_ABS**
- **DsHrConformantIDL_1_0_Mgr**
- **DsHrConformantProxy** -- Provides a client interface to the server's distributed object..

- **DsHrDCEConformantConcrete** -- Encapsulates all DCE exposure for the connection objects on the server side.
- **DsHrDCENonConfConcrete** -- Encapsulates all DCE exposure for the connection objects on the server side.
- **DsHrEnv** -- This class provides the ability to check and get environment variables that are required buy HDF operations.
- **DsHrEvent --** Declaration of event class.
- **DsHrFhExecutor** -- This is Fault handler class for DsHrConformantProxy and DsHrNonConfProxy.
- **DsHrHdfEosListener** -- This class is to compare the current memory size with the previous and according to some established policy, restart the HdfEos server or shut down the HdfEos server.
- **DsHrListener** -- The class is an abstract class which is instanciated by its derived class - DsHrHdfEosListener.
- **DsHrManagedServer** -- This class is dervived from EcPfManagedServer and invoked by HDFEOS server.
- **DsHrNonConfIDL_1_0**
- **DsHrNonConfIDL_1_0_ABS**
- **DsHrNonConfIDL_1_0_Mgr**
- **DsHrNonConfProxy** -- Provides a client interface to the server's distributed object..
- **DsHrQuit** -- This class will no longer be needed when the MSS lifecycle services and/or the managed process framework are complete.
- **DsHrQuitIDL_1_0**
- **DsHrQuitIDL_1_0_ABS**
- **DsHrQuitIDL_1_0_Mgr**
- **DsHrRestartManager** -- Declaration of restart manager class.
- **DsHrTools** -- Provides common functions needed by both Proxy and Concrete.

library DsMd1Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

Classes

- **DsMdAttributeList** -- This singleton class stores the description of a stored procedure or dynamic SQL statement associated with Insert, Update, or Retrieval of a GlParameterList to the database.
- **DsMdCatalogBase** -- This class is used to write entities to the Sybase database.
- **DsMdCollDictionary** -- This singleton class is used to cache the collection level metadata for ESDTs stored within the SDSRV.

- **DsMdID** -- This class serves as a wrapper class around the database ID.
- **DsMdJoinTable** -- This singleton class is used to store all the possible joins between the various tables in the database.
- **DsMdPSTableAliasList** -- This class is used to store the FROM clause elements associated with search for Product Specific Attributes within the database.

library DsMd2Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

Classes

- **DsMdCatalog** -- This class controls the metadata transations: Insert, Update, Retrieve, and Search.
- **DsMdCollDictionary** -- This singleton class is used to cache the collection level metadata for ESDTs stored within the SDSRV.
- **DsMdESDTConstraint**
- **DsMdESDTConstraintArray**
- **DsMdGRingPoint** -- This class supports GPolygon metadata insert by providing a class whose objects can be sorted by the "SequenceNumber" of the point.
- **DsMdMetadata** -- This class serves as a wrapper around the GlParameterList containing the metadata attibutes.
- **DsMdODL** -- This class holds an ODL structure from an external file in memory and provides interfaces to manipulate this structure.

library DsMd3Sh (shared)

/ecs/formal/DSS/sdsrv/src/md

Classes

- **DsMdAttr** -- This class is not currently used by the system but is intended to help with porting from Sybase to other database systems.
- **DsMdBaseDep** -- Contains the list of tables to be processed as part of a metadata DeletePhysical.
- **DsMdCommand** -- Supports DeletePhysical by translating logical action names and table names to the corresponding Stored Procedure to execute.
- **DsMdConvertToFromSQL** -- This class contains methods which implement the LogicalDelete, LogicalUnDelete, and DeletePhysical functionalities
- **DsMdInsertSeq** -- Contains the list of metadata lists to be processed as part of a metadata Insert.
- **DsMdLookuptableBase** -- A base class and derived singleton classes for maintaining lookup tables initially loaded from the database.

- **DsMdOperator** -- Translates from a logical operator name to the actual physical string required by the DBMS.
- **DsMdUnloadList** -- Contains the list of metadata lists to be processed as part of a metadata Retrieval.
- **DsMdUpdateXrefList** -- Contains the list of metadata lists to be processed as part of a metadata Update.

library DsSh

/ecs/formal/DSS/sdsrv/src/sh

This library provides underlying DCE(OODCE) mechanism and fundamental implementation for both client and server side of SDSRV. It also supplies utilities which is used by both client and server side code of SDSRV, such as cl, sr, ge, md, de libraries and etc. Some of its facilities are also used by other subsystems.

Classes

- **DsShCompression** -- Defines the type of compression.
- **DsShConnectionFhExecutor** -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- **DsShConnectionIDL_4_0**
- **DsShConnectionIDL_4_0_ABS**
- **DsShConnectionIDL_4_0_Mgr**
- **DsShConnectionProxy** -- This class provides a client interface to the server's connection distributed object.
- **DsShConnectionReal** -- This class provides a server interface to the server's connection distributed object. It inherits from the DCE-generated server connection class, and adds functions to provide stronger type-checking.
- **DsShConVar** -- Provides a condition variable class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) condition variable implementation.
- **DsShDCEConnectionConcrete** -- This class encapsulates all DCE exposure for the connection objects on the server-side.
- **DsShDCEDescriptorConcrete** -- This class encapsulates all of the DCE behavior and signatures for Descriptors and Descriptor collectors.
- **DsShDCEId** -- This class encapsulates the DCE object identifier, uuid_t.
- **DsShDCEInterface** -- This class encapsultes DCE implementations of DO interface functions, and adds other functions needed by other DCE implementation objects.
- **DsShDCERef** -- This class encapsulates the DCE object reference type DCEObjRefT.

- **DsShDCESRequestConcrete** -- This server-side class encapsulates all of the DCE behavior and signatures for srequests, and communicates with an associated DsShSRequestReal in a "DCE-clean" manner.
- **DsShDCEUpdaterConcrete** -- This class encapsulates all DCE exposure for the Updater objects on the server-side.
- **DsShDescriptorFhExecutor** -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- **DsShDescriptorIDL_4_0**
- **DsShDescriptorIDL_4_0_ABS**
- **DsShDescriptorIDL_4_0_Mgr**
- **DsShDescriptorProxy** -- This class provides a DCE-independent interface to the Descriptor object and to the set of descriptor objects.
- **DsShDescriptorReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsShDOId** -- Declaration of a distributed object (DO) id, independent of the underlying technology.
- **DsShDOInterface** -- This class provides a basic interface that all distributed object (DO) technologies must supply for the dataserver.
- **DsShDORef** -- This class encapsulates references to distributed objects in a manner that is independent of the underlying distributed object (DO) technology.
- **DsShError** -- This class is used for exceptions and error reporting in dataserver functions.
- **DsShESDTProvider** -- This class is a shell provider for the ESDT UR.
- **DsShESDTUR** -- A UR is a persistent, transportable identifier that refers to C++ objecs.
- **DsShGenCommandImp** -- Implementation class for commands.
- **DsShGenRequestImp** -- Implementation class for requests.
- **DsShGenRequestMediator** -- A RequestMediator object will mediate interactions between DsShDCEConnectionConcrete (CC), DsShDCESRequestConcrete (RC), and DsShSRequestReal (RR) objects.
- **DsShGenSessionManager** -- The management of Sessions in the Sdsrv server should go into this class, inlcuding The creation and destruction of Sessions and accounting information.
- **DsShManagedClient** -- DSS client which uses Process Frame Work .
- **DsShMutex** -- This class provides a mutex class that encapsulates the underlying (currently DCE) mutex implementation.
- **DsShMutexLock** -- This class provides a safe method of locking and unlocking DsShMutex objects, especially in the presence of exceptions.

- **DsShNoPFTestPro** -- This is a the declaration of DsShNoPFTestPro Class, which is solely for Test programs which do not have a PF initiated.
- **DsShODLMutex** -- This mutex will protect access to single thread ODL lib.
- **DsShQuitIDL_4_0**
- **DsShQuitIDL_4_0_ABS**
- **DsShQuitIDL_4_0_Mgr**
- **DsShQuitProxy** -- Client interface to the server shutdown class.
- **DsShQuitReal** -- This is a server-side class that implements code to shutdown the dataserver.
- **DsShSciCommandImp** -- This class specializes DsShGenCommandImp for the science data server to provide command categories.
- **DsShSciRequestImp** -- This class specializes DsShGenRequestImp for the science server, adding priorities and domains.
- **DsShSciServerUR**
- **DsShServerUR** -- Generic Server UR.
- **DsShServerURProvider** -- Provides URs to clients.
- **DsShSRequestIDL_4_0**
- **DsShSRequestIDL_4_0_ABS**
- **DsShSRequestIDL_4_0_Mgr**
- **DsShSRequestProxy** -- This class provides a client interface to the server's request distributed object, and translates the data types into the required DCE formats before calling the underlying DCE client-side request object..
- **DsShSRequestReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking..
- **DsShThread** -- This class provides a thread class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) thread implementation.
- **DsShTypeID** -- This class represents a unique ESDT.
- **DsShUpdaterIDL_4_0**
- **DsShUpdaterIDL_4_0_ABS**
- **DsShUpdaterIDL_4_0_Mgr**
- **DsShUpdaterProxy** -- This class provides a client interface to the server's Updater distributed object.
- **DsShUpdaterReal** -- This class provides a server interface to the server's updater distributed object. It is primarily used in data server rundown condition while the client is making a asynchronous request with call back..

library DsShSh (shared)

/ecs/formal/DSS/sdsrv/src/sh

This internal library provides underlying DCE(OODCE) mechanism and fundamental implementation for both client and server side of SDSRV. It also supplies utilities which is used by both client and server side code of SDSRV, such as cl, sr, ge, md, de libraries and etc. The library has both static and shared versions. (unfortunately, due to historical reasons. some DsSh facilities are also used by other subsystems)

Classes

- **DsShCompression** -- Defines the type of compression.
- **DsShConnectionFhExecutor** -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- **DsShConnectionIDL_4_0**
- **DsShConnectionIDL_4_0_ABS**
- **DsShConnectionIDL_4_0_Mgr**
- **DsShConnectionProxy** -- This class provides a client interface to the server's connection distributed object.
- **DsShConnectionReal** -- This class provides a server interface to the server's connection distributed object. It inherits from the DCE-generated server connection class, and adds functions to provide stronger type-checking.
- **DsShConVar** -- Provides a condition variable class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) condition variable implementation.
- **DsShDCEConnectionConcrete** -- This class encapsulates all DCE exposure for the connection objects on the server-side.
- **DsShDCEDescriptorConcrete** -- This class encapsulates all of the DCE behavior and signatures for Descriptors and Descriptor collectors.
- **DsShDCEId** -- This class encapsulates the DCE object identifier, uuid_t.
- **DsShDCEInterface** -- This class encapsultes DCE implementations of DO interface functions, and adds other functions needed by other DCE implementation objects.
- **DsShDCERef** -- This class encapsulates the DCE object reference type DCEObjRefT.
- **DsShDCESRequestConcrete** -- This server-side class encapsulates all of the DCE behavior and signatures for srequests, and communicates with an associated DsShSRequestReal in a "DCE-clean" manner.
- **DsShDCEUpdaterConcrete** -- This class encapsulates all DCE exposure for the Updater objects on the server-side.
- **DsShDescriptorFhExecutor** -- This is Fault handler class used for Submit() method in DsShConnectionProxy class.
- **DsShDescriptorIDL_4_0**

- **DsShDescriptorIDL_4_0_ABS**
- **DsShDescriptorIDL_4_0_Mgr**
- **DsShDescriptorProxy** -- This class provides a DCE-independent interface to the Descriptor object and to the set of descriptor objects.
- **DsShDescriptorReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsShDOId** -- Declaration of a distributed object (DO) id, independent of the underlying technology.
- **DsShDOInterface** -- This class provides a basic interface that all distributed object (DO) technologies must supply for the dataserver.
- **DsShDORef** -- This class encapsulates references to distributed objects in a manner that is independent of the underlying distributed object (DO) technology.
- **DsShError** -- This class is used for exceptions and error reporting in dataserver functions.
- **DsShESDTProvider** -- This class is a shell provider for the ESDT UR.
- **DsShESDTUR** -- A UR is a persistent, transportable identifier that refers to C++ objecs.
- **DsShGenCommandImp** -- Implementation class for commands.
- **DsShGenRequestImp** -- Implementation class for requests.
- **DsShGenRequestMediator** -- A RequestMediator object will mediate interactions between DsShDCEConnectionConcrete (CC), DsShDCESRequestConcrete (RC), and DsShSRequestReal (RR) objects.
- **DsShGenSessionManager** -- The management of Sessions in the Sdsrv server should go into this class, inlcuding The creation and destruction of Sessions and accounting information.
- **DsShManagedClient** -- DSS client which uses Process Frame Work .
- **DsShMutex** -- This class provides a mutex class that encapsulates the underlying (currently DCE) mutex implementation.
- **DsShMutexLock** -- This class provides a safe method of locking and unlocking DsShMutex objects, especially in the presence of exceptions.
- **DsShNoPFTestPro** -- This is a the declaration of DsShNoPFTestPro Class, which is solely for Test programs which do not have a PF initiated.
- **DsShODLMutex** -- This mutex will protect access to single thread ODL lib.
- **DsShQuitIDL_4_0**
- **DsShQuitIDL_4_0_ABS**
- **DsShQuitIDL_4_0_Mgr**
- **DsShQuitProxy** -- Client interface to the server shutdown class.

- **DsShQuitReal** -- This is a server-side class that implements code to shutdown the dataserver.
- **DsShSciCommandImp** -- This class specializes DsShGenCommandImp for the science data server to provide command categories.
- **DsShSciRequestImp** -- This class specializes DsShGenRequestImp for the science server, adding priorities and domains.
- **DsShSciServerUR**
- **DsShServerUR** -- Generic Server UR.
- **DsShServerURProvider** -- Provides URs to clients.
- **DsShSRequestIDL_4_0**
- **DsShSRequestIDL_4_0_ABS**
- **DsShSRequestIDL_4_0_Mgr**
- **DsShSRequestProxy** -- This class provides a client interface to the server's request distributed object, and translates the data types into the required DCE formats before calling the underlying DCE client-side request object..
- **DsShSRequestReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking..
- **DsShThread** -- This class provides a thread class that encapsulates (and provides roughly the same functionality as) the underlying (currently DCE/pthreads) thread implementation.
- **DsShTypeID** -- This class represents a unique ESDT.
- **DsShUpdaterIDL_4_0**
- **DsShUpdaterIDL_4_0_ABS**
- **DsShUpdaterIDL_4_0_Mgr**
- **DsShUpdaterProxy** -- This class provides a client interface to the server's Updater distributed object.
- **DsShUpdaterReal** -- This class provides a server interface to the server's updater distributed object. It is primarily used in data server rundown condition while the client is making a asynchronous request with call back..

library DsSiSh (shared)

/ecs/formal/DSS/sdsrv/src/si

Classes

- **DsSiMemento** -- et a catalog from the pool call DsSiSysCatalog::DeleteMemento(DsSiMemento* pMemento) then recursiveDelete on myStateAttributes then initializes all the other attributes.

- **DsSiMemMonitor** -- This class is used to reserve and monitor free memory conditions within the process.
- **DsSiSysDataCatalog**

library DsSr

/ecs/formal/DSS/sdsrv/src/sr

This is an internal library for Science Data Server. It provides some functions which are used by other directories's programs. These functions include reading configuration file, get access to metadata database and get access to HDFEOS server and so on.

Classes

- **DsSrConnectionMaker** -- Provides a specialized NewConnection function for DsShDCEConnectionConcrete to create DsSrSession objects.
- **DsSrDescriptorMaker** -- Provides specialized services for creation of server side descriptor objects.
- **DsSrEnv** -- This class provides the ability to check and get enviroment variables that are required buy DSS operations.
- **DsSrGenCatalogPool** -- Used by Science Data Server objects to obtain access to a catalog.
- **DsSrGenCatalogWrapper** -- Used by data server objects to obtain access to a catalog.
- **DsSrGenConnection** -- Distributed base class for all connections/sessions that exist in Science Data Server.
- **DsSrGenQueuedConnection** -- Provides the ability to queue requests as they arrive.
- **DsSrGenRequestInt** -- Server side object that represents a request to the data server.
- **DsSrHDFServerPool** -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- **DsSrHDFServerToken** -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- **DsSrManagedServer** -- Hides the interface between MSS and the application by instantiaing the EcAgManager object.
- **DsSrRequest** -- Server side object that represents a request to the Science Data Server.
- **DsSrSession** -- A specialization of the DsSrGenQueuedConnection class that adds DsSrWorkingCollection-oriented behavior to the connection.
- **DsSrSysCatalogPool** -- Used by Science Data Server objects to obtain access to a System Catalog.
- **DsSrSysCatalogWrapper** -- Used by Science Data Server objects to obtain access to a System Catalog.
- **DsSrWarmStartManager**

- **DsSrWorkingCollection** -- Maintains a vector of ESDT objects that represents the session's current contents..

library DsSrSh (shared)

/ecs/formal/DSS/sdsrv/src/sr

This is an internal library for Science Data Server. It provides some functions which are used by other directories's programs. These functions include reading configuration file, get access to metadata database and get access to HDFEOS server and so on.

Classes

- **DsSrConnectionMaker** -- Provides a specialized NewConnection function for DsShDCEConnectionConcrete to create DsSrSession objects.
- **DsSrDescriptorMaker** -- Provides specialized services for creation of server side descriptor objects.
- **DsSrEnv** -- This class provides the ability to check and get enviroment variables that are required buy DSS operations.
- **DsSrGenCatalogPool** -- Used by Science Data Server objects to obtain access to a catalog.
- **DsSrGenCatalogWrapper** -- Used by data server objects to obtain access to a catalog.
- **DsSrGenConnection** -- Distributed base class for all connections/sessions that exist in Science Data Server.
- **DsSrGenQueuedConnection** -- Provides the ability to queue requests as they arrive.
- **DsSrGenRequestInt** -- Server side object that represents a request to the data server.
- **DsSrHDFServerPool** -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- **DsSrHDFServerToken** -- Used by the Science Data Server to obtain access to the HDF EOS Servers.
- **DsSrManagedServer** -- Hides the interface between MSS and the application by instantiaing the EcAgManager object.
- **DsSrRequest** -- Server side object that represents a request to the Science Data Server.
- **DsSrSession** -- A specialization of the DsSrGenQueuedConnection class that adds DsSrWorkingCollection-oriented behavior to the connection.
- **DsSrSysCatalogPool** -- Used by Science Data Server objects to obtain access to a System Catalog.
- **DsSrSysCatalogWrapper** -- Used by Science Data Server objects to obtain access to a System Catalog.
- **DsSrWarmStartManager**
- **DsSrWorkingCollection** -- Maintains a vector of ESDT objects that represents the session's current contents..

library DsSt8mmTapeShSh (shared)

/ecs/formal/DSS/stmgt/src/tp/8mm

Classes

- **DsSt8mmTape**

library DsSt9TrkTapeShSh (shared)

/ecs/formal/DSS/stmgt/src/tp/9Trk

Classes

- **DsSt9TrkTape**

library DsStCDROMShSh (shared)

/ecs/formal/DSS/stmgt/src/cd

Classes

- **DsStCDROM**
- **DsStCDROMImp**
- **DsStCDROMResourceConfig**

library DsStCmn

/ecs/formal/DSS/stmgt/src/cmn

This internal library contains the interface that the code uses to do database access and run stored procedures. It also contains the shared code used by all servers for basic server management.

Classes

- **DsStBaseCFhExecutor** -- All custom stmgt executors will inherit from this class.
- **DsStCommonDBIF** -- If a global instance is needed which is outside of a class declare the variable static.
- **DsStDbConnection**
- **DsStDbConnectionManager**
- **DsStDbInterface** -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.
- **DsStDBInterface** -- This class is used to initialize the database and the database server.
- **DsStDCEServer**
- **DsStDCEStreamer**
- **DsStDictionary** -- Data dictionary that maps a string name to a string value.
- **DsStDirParse**

- **DsStErrorDetails** -- DsStErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- **DsStFtpWrappers** -- DsStFtpWrappers::DsStFtpWrappers(const DsStFtpWrappers&).
- **DsStLogger** -- Class which logs all DSS messages to a log.
- **DsStParameter** -- Represents a value of a parameter.
- **DsStParameterList** -- DsStParameterList class declaration.
- **DsStServerConcrete** -- Declarations for DsStServerConcrete class which is a pure virtual base class used to invoke the Startup() method.
- **DsStStoredProcedures**
- **DsStStProcTable** -- This is a singleton used to create a dictionary that associates a stored procedure name with a list of parameters in the order that they appear in the stored procedure declaration.
- **DsStTestDriver**
- **DsStUtility**

library DsStCmnShSh (shared)

/ecs/formal/DSS/stmgt/src/cmn

Classes

- **DsStBaseCFhExecutor** -- All custom stmgt executors will inherit from this class.
- **DsStCommonDBIF** -- If a global instance is needed which is outside of a class declare the variable static.
- **DsStDbConnection**
- **DsStDbConnectionManager**
- **DsStDbInterface** -- Database (Sybase) interface class to encapsulate database related services such as: connect, execute, fetch result.
- **DsStDBInterface** -- This class is used to initialize the database and the database server.
- **DsStDCEServer**
- **DsStDCEStreamer**
- **DsStDictionary** -- Data dictionary that maps a string name to a string value.
- **DsStDirParse**
- **DsStErrorDetails** -- DsStErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- **DsStFtpWrappers** -- DsStFtpWrappers::DsStFtpWrappers(const DsStFtpWrappers&).
- **DsStLogger** -- Class which logs all DSS messages to a log.
- **DsStParameter** -- Represents a value of a parameter.

- **DsStParameterList** -- DsStParameterList class declaration.
- **DsStServerConcrete** -- Declarations for DsStServerConcrete class which is a pure virtual base class used to invoke the Startup() method.
- **DsStStoredProcedures**
- **DsStStProcTable** -- This is a singleton used to create a dictionary that associates a stored procedure name with a list of parameters in the order that they appear in the stored procedure declaration.
- **DsStTestDriver**
- **DsStUtility**

library DsStD3Tape

/ecs/formal/DSS/stmgt/src/tp/D3

This is a server side library only and contains the custom code for support of the D3 tape. It is only supported on the Sun.

Classes

- **DsStD3Tape**

library DsStDb

/ecs/formal/DSS/stmgt/src/tp/stacker/db

This library contains the implementation of the common stacker code. It is supported only for the SUN architecture and should only be used by the tape stacker resource managers on the server side.

Classes

- **DsStDriveListEntry** -- This class is a RWCollectable so that it can be added in a list as a node.
- **DsStDriveListTable** -- This class serves as an encapsulation of the Drive table in the database where it resides.
- **DsStSlotListEntry** -- This class is a RWCollectable so that it can be put in a list as a node.
- **DsStSlotListTable** -- This class serves as an encapsulation of the Slot table in the database context where this table exists.
- **DsStStacker** -- The class inherits from DsStStackerListEntry because it needs all the data members of a stacker record from the stacker table and it will add the SCSI device stacker operations.
- **DsStStackerConfig** -- This class is used by DsStStacker to read the stacker configuration parameters from a config file.
- **DsStStackerIF** -- StackerIF - DataBase wrapper object.

- **DsStStackerListEntry** -- This class is a RWCollectable so that it can be part of a list as a node.
- **DsStStackerListTable** -- This class serves as an encapsulation of a table in the database containing the Stacker table.
- **DsStStackerManager** -- This class is a singleton which contains the resource allocation/deallocation for Ingest/Distribution and the the request queuing/dequeuing algorithms.
- **DsStStackerQueue** -- The DsStStackerQueue (the Queue) holds stacker ingest or distribution requests that are waiting to be processed.
- **DsStStackerQueueEntry** -- This class encapsulates the object which will be placed in the StackerQueue as a result of an Ingest or Distribution allocation than can't be serviced immediately..

library DsStDisFtp

/ecs/formal/DSS/stmgt/src/ftp/disftp

This library contains the server code which allows for the moving of data from staging to the target location either for end user pull of data from the pull area or push of data to the end user specified node and directory.

Classes

- **DsStDistributionFtp**

library DsStDisFtpShSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/disftp

Classes

- **DsStDistributionFtp**

library DsStFmt

/ecs/formal/DSS/stmgt/src/tp/fmt

This library is used to format the writing of files to tape and other reading of files from tape. For Version 2.0 the only supported format is tar. It is supported only for the Sun.

Classes

- **DsStFormatCreator**
- **DsStTarFormat**

library DsStFmtShSh (shared)

/ecs/formal/DSS/stmgt/src/tp/fmt

Classes

- **DsStFormatCreator**
- **DsStTarFormat**

library DsStFtp

/ecs/formal/DSS/stmgt/src/ftp/ftpclient

Classes

- **FtpLibrary** -- Library for ftpd clients.(libftp).

library DsStFtpNotifyShSh (shared)

/ecs/formal/DSS/stmgt/src/FtpNotify

Classes

- **DsStFtpNotifyManager**
- **DsStSyslogOutputFile**

library DsStFtpSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/ftpclient

Classes

- **FtpLibrary** -- Library for ftpd clients.(libftp).

library DsStGuiCmn

/ecs/formal/DSS/gui/src/cmn

Classes

- **DsGuBusyCursor** -- The constructor and destructor are always called in pairs to set/reset the cursor to indicate that the application is busy and unable to process user input in a window.
- **DsGuErrorDialog**
- **DsGuExtendedList** -- This class provides utility routines for use with the BX extended list widget.
- **DsGuIntegerControl**
- **DsGuOperatorMessage**
- **DsGuRealControl**
- **DsGuStreetLight**
- **UIComponent** -- Base class for all C++/Motif UI components.

library DsStIngestFtp

/ecs/formal/DSS/stmgt/src/ftp/inftp

This library contains the server side code for performing ftp pulls to satisfy Ingest. It also requires the CSS library for ftp services.

Classes

- **DsStIngestFtp**

library DsStIngestFtpShSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/inftp

Classes

- **DsStIngestFtp**

library DsStNet

/ecs/formal/DSS/stmgt/src/ftp/net

This is a library for the server side only and contains the base class functionality for the classes which implement ftp push and ftp pull distribution services.

Classes

- **DsStFtpResourceConfig**
- **DsStNetworkResource** -- The DsStNetwork Resource class has common functionality for any NetworkResource.

library DsStNetShSh (shared)

/ecs/formal/DSS/stmgt/src/ftp/net

Classes

- **DsStFtpResourceConfig**
- **DsStNetworkResource** -- The DsStNetwork Resource class has common functionality for any NetworkResource.

library DsStPrinterShSh (shared)

/ecs/formal/DSS/stmgt/src/pr

Classes

- **DsStPrinter**
- **DsStPrinterResourceConfig**
- **DsStPrintPackingSlips**

library DsStPull

/ecs/formal/DSS/stmgt/src/pull

This library contains the server and client code for the pull cache manager. It is supported on both Sun and SGI hardware.

Classes

- **DsStDCEPullMonitorConcrete** -- Class which monitors the space usage of the pull area.
- **DsStPMIDL_4_0**
- **DsStPMIDL_4_0_ABS**
- **DsStPMIDL_4_0_Mgr**
- **DsStPullCacheFileDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the Cache File table.
- **DsStPullFileLocDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the File Location table.
- **DsStPullFtpNotify**
- **DsStPullMonitor** -- Class which monitors the space usage of the pull area.
- **DsStPullMonitorCFhExecutor**
- **DsStPullMonitorIF**
- **DsStPullMonitorProxy** -- Class which monitors the space usage of the pull area.
- **DsStPullMonitorReal** -- Class which monitors the space usage of the pull area.

library DsStPullShSh (shared)

/ecs/formal/DSS/stmgt/src/pull

Classes

- **DsStDCEPullMonitorConcrete** -- Class which monitors the space usage of the pull area.
- **DsStPMIDL_4_0**
- **DsStPMIDL_4_0_ABS**
- **DsStPMIDL_4_0_Mgr**
- **DsStPullCacheFileDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the Cache File table.
- **DsStPullFileLocDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the File Location table.
- **DsStPullFtpNotify**
- **DsStPullMonitor** -- Class which monitors the space usage of the pull area.
- **DsStPullMonitorCFhExecutor**

- **DsStPullMonitorIF**
- **DsStPullMonitorProxy** -- Class which monitors the space usage of the pull area.
- **DsStPullMonitorReal** -- Class which monitors the space usage of the pull area.

library DsStRes

/ecs/formal/DSS/stmgt/src/rc

This library provides the common functionality for all of the resource manager servers.Libraries are maintained for both SGI and Sun. SGI versions are used in support of ftp services. Sun versions are used in support of hard media distribution. Both client and server side code reside in this library.

Classes

- **DsStDCEResourceProviderConcrete** -- This server-side class encapsulates all of the behavior and signatures.
- **DsStPhysicalResource**
- **DsStPhysicalResourceConfig**
- **DsStRequestIF**
- **DsStRequestListEntry**
- **DsStRequestListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.
- **DsStRequestManager** -- DsStRequestManager manages requests.
- **DsStReserListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.
- **DsStReservation**
- **DsStReservationListEntry** -- This class is a RWCollectable so that it can be put on a list as the node.
- **DsStResource** -- DsStResource is the base class for all resource objects.
- **DsStResourceConfig** -- Retrieves information from configuration file.
- **DsStResourceIF**
- **DsStResourceListEntry**
- **DsStResourceManager** -- The DsStResourceManager manages the resources.
- **DsStResourcePolicy**
- **DsStResourceProvider** -- he user of this method is responsible for check if the pointer returned is not NULL before utilizing it.
- **DsStResourceProviderCFhExecutor**
- **DsStResourceProviderProxy** -- This class provides a client interface to the server's descriptor distributed object..

- **DsStResourceProviderReal** -- It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsStResourceQueue** -- The DsStResourceQueue (the Queue) holds requests that are waiting to be processed.
- **DsStResourceSchedule** -- The DsStResourceSchedule schedule the resources.
- **DsStResProvIDL_4_0**
- **DsStResProvIDL_4_0_ABS**
- **DsStResProvIDL_4_0_Mgr**
- **DsStSchedularListEntry**
- **DsStSchedularListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.

library DsStResShSh (shared)

/ecs/formal/DSS/stmgt/src/rc

Classes

- **DsStDCEResourceProviderConcrete** -- This server-side class encapsulates all of the behavior and signatures.
- **DsStPhysicalResource**
- **DsStPhysicalResourceConfig**
- **DsStRequestIF**
- **DsStRequestListEntry**
- **DsStRequestListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.
- **DsStRequestManager** -- DsStRequestManager manages requests.
- **DsStReserListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.
- **DsStReservation**
- **DsStReservationListEntry** -- This class is a RWCollectable so that it can be put on a list as the node.
- **DsStResource** -- DsStResource is the base class for all resource objects.
- **DsStResourceConfig** -- Retrieves information from configuration file.
- **DsStResourceIF**
- **DsStResourceListEntry**
- **DsStResourceManager** -- The DsStResourceManager manages the resources.
- **DsStResourcePolicy**

- **DsStResourceProvider** -- he user of this method is responsible for check if the pointer returned is not NULL before utilizing it.
- **DsStResourceProviderCFhExecutor**
- **DsStResourceProviderProxy** -- This class provides a client interface to the server's descriptor distributed object..
- **DsStResourceProviderReal** -- It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsStResourceQueue** -- The DsStResourceQueue (the Queue) holds requests that are waiting to be processed.
- **DsStResourceSchedule** -- The DsStResourceSchedule schedule the resources.
- **DsStResProvIDL_4_0**
- **DsStResProvIDL_4_0_ABS**
- **DsStResProvIDL_4_0_Mgr**
- **DsStSchedularListEntry**
- **DsStSchedularListTable** -- This class serves as an encapsulation of a table in the database containing entries in the resource area.

library DsStScsi

/ecs/formal/DSS/stmgt/src/scsi

This library supports most standard scsi commands in hopes that there will only be minimal changes needed to support new devices. It is supported only on the Sun.

Classes

- **DsStSCSI** -- This is the base class for all SCSI interface classes.
- **DsStSCSIImp** -- Class Defintion.
- **DsStSCSILibrary**
- **DsStSCSILibraryEXB210** -- SCSI Library Commands.
- **DsStSCSITape** -- This class, which is derived from DsStSCSI, is the base class for all types of SCSI tape interface.
- **DsStSUNSCSIImp** -- This class provides the SUN specific User SCSI interface.

library DsStScsiShSh (shared)

/ecs/formal/DSS/stmgt/src/scsi

Classes

- **DsStSCSI** -- This is the base class for all SCSI interface classes.
- **DsStSCSIImp** -- Class Defintion.

- **DsStSCSILibrary**
- **DsStSCSILibraryEXB210** -- SCSI Library Commands.
- **DsStSCSITape** -- This class, which is derived from DsStSCSI, is the base class for all types of SCSI tape interface.
- **DsStSUNSCSIImp** -- This class provides the SUN specific User SCSI interface.

library DsStSt

/ecs/formal/DSS/stmgt/src/st

Both client and server code are contained in the library. In addition to the server code, theere is also a utilities class which is used by the different servers to perform some share functionality.

Classes

- **DsStArchive**
- **DsStArchiveCFhExecutor**
- **DsStArchiveIDL_4_0**
- **DsStArchiveIDL_4_0_ABS**
- **DsStArchiveIDL_4_0_Mgr**
- **DsStArchiveProxy** -- This class provides a client interface to the server's descriptor distributed object.
- **DsStArchiveReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsStCompressCreator** -- This class is to return the object of the type of compression requested.
- **DsStDCEArchiveConcrete**
- **DsStDCEStagingDiskConcrete**
- **DsStDCEStagingMonitorConcrete**
- **DsStDiskRequestManager** -- DsStDiskRequestManager manages requests.
- **DsStFileAttributes** -- imeStamp (EcTiTimeService time);.
- **DsStFileCompress** -- This class is to compress files.
- **DsStFileInfo**
- **DsStFileParameters** -- Provides basic CSDT functionality.
- **DsStStageDiskIDL_4_0**
- **DsStStageDiskIDL_4_0_ABS**
- **DsStStageDiskIDL_4_0_Mgr**

- **DsStStagingCacheFileDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the read only Cache File table.
- **DsStStagingDataList** -- o attributes.
- **DsStStagingDisk** -- DsStStagingDisk class allows access to files and directories on the working storage disk space managed by the storage management software.
- **DsStStagingDiskCFhExecutor**
- **DsStStagingDiskController** -- This class allows another class to update the Staging Disk remaining space and have access to the path for the staging Disk Users of this class take explicit responsibility for the updates to the size of the staging disk.
- **DsStStagingDiskProxy** -- This class provides a layer between the implementation of the DsStStagingDisk class and the underlying network protocol to make the DsStStagingDisk act as a distributed class
- **DsStStagingDiskReal**
- **DsStStagingManager**
- **DsStStagingMonitor**
- **DsStStagingMonitorCFhExecutor**
- **DsStStagingMonitorIF**
- **DsStStagingMonitorProxy**
- **DsStStagingMonitorReal**
- **DsStStagingMonitorRequestQueueDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the Staging Monitor Request Queue File table.
- **DsStStagingQueue** -- The DsStStagingQueue (the Queue) holds requests that are waiting to be processed.
- **DsStStagingQueueEntry** -- This class represent the entries in the staging table for each staging disk that is completed.
- **DsStStagingTable** -- Contains the class definition for the DsStStagingTable Class.
- **DsStStagingTableEntry** -- This class represent the entries in the staging table for each staging disk that is completed.
- **DsStStgMonIDL_4_0**
- **DsStStgMonIDL_4_0_ABS**
- **DsStStgMonIDL_4_0_Mgr**
- **DsStUnixCompress** -- This class is used to do Unix compression.

library DsStStk4

/ecs/formal/DSS/stmgt/src/tp/stacker/4mm

This library implements code specific for the 4MM stacker resource manager and should only be used by the server side. Client interface to all the resource managers is in the libDsStRes library.

Classes

- **DsSt4mmStackerTape** -- This class will provide the 4mmTape functionality for the Tape Drive belonging to a 4mm stacker.

library DsStStk8

/ecs/formal/DSS/stmgt/src/tp/stacker/8mm

This library implements code specific for the 8MM stacker resource manager and should only be used by the server side. Client interface to all the resource managers is in the libDsStRes library. It is supported only on the Sun.

Classes

- **DsSt8mmStackerTape** -- This class will provide the 8mmTape functionality for the Tape Drive belonging to a 8mm stacker.

library DsStStSh (shared)

/ecs/formal/DSS/stmgt/src/st

Both client and server code are contained in this library. In addition to the server code, theere is also a utilities class which is used by the different servers to perform some share functionality.

Classes

- **DsStArchive**
- **DsStArchiveCFhExecutor**
- **DsStArchiveIDL_4_0**
- **DsStArchiveIDL_4_0_ABS**
- **DsStArchiveIDL_4_0_Mgr**
- **DsStArchiveProxy** -- This class provides a client interface to the server's descriptor distributed object.
- **DsStArchiveReal** -- This class provides a server interface to the server's request distributed object. It inherits from the DCE-generated server request class, and adds functions to provide stronger type-checking.
- **DsStCompressCreator** -- This class is to return the object of the type of compression requested.
- **DsStDCEArchiveConcrete**
- **DsStDCEStagingDiskConcrete**
- **DsStDCEStagingMonitorConcrete**
- **DsStDiskRequestManager** -- DsStDiskRequestManager manages requests.
- **DsStFileAttributes** -- imeStamp (EcTiTimeService time);.
- **DsStFileCompress** -- This class is to compress files.

- **DsStFileInfo**
- **DsStFileParameters** -- Provides basic CSDT functionality.
- **DsStStageDiskIDL_4_0**
- **DsStStageDiskIDL_4_0_ABS**
- **DsStStageDiskIDL_4_0_Mgr**
- **DsStStagingCacheFileDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the read only Cache File table.
- **DsStStagingDataList** -- o attributes.
- **DsStStagingDisk** -- DsStStagingDisk class allows access to files and directories on the working storage disk space managed by the storage management software.
- **DsStStagingDiskCFhExecutor**
- **DsStStagingDiskController** -- This class allows another class to update the Staging Disk remaining space and have access to the path for the staging Disk Users of this class take explicit responsibility for the updates to the size of the staging disk.
- **DsStStagingDiskProxy** -- This class provides a layer between the implementation of the DsStStagingDisk class and the underlying network protocol to make the DsStStagingDisk act as a distributed class
- **DsStStagingDiskReal**
- **DsStStagingManager**
- **DsStStagingMonitor**
- **DsStStagingMonitorCFhExecutor**
- **DsStStagingMonitorIF**
- **DsStStagingMonitorProxy**
- **DsStStagingMonitorReal**
- **DsStStagingMonitorRequestQueueDBIF** -- This class interfaces with DsStCommonDBIF to perform DB queries on the Staging Monitor Request Queue File table.
- **DsStStagingQueue** -- The DsStStagingQueue (the Queue) holds requests that are waiting to be processed.
- **DsStStagingQueueEntry** -- This class represent the entries in the staging table for each staging disk that is completed.
- **DsStStagingTable** -- Contains the class definition for the DsStStagingTable Class.
- **DsStStagingTableEntry** -- This class represent the entries in the staging table for each staging disk that is completed.
- **DsStStgMonIDL_4_0**
- **DsStStgMonIDL_4_0_ABS**
- **DsStStgMonIDL_4_0_Mgr**

- **DsStUnixCompress** -- This class is used to do Unix compression.

library DsStTape

/ecs/formal/DSS/stmgt/src/tp/tape

This library contains the base class functionality that is common to all tape resource managers. This libary should only be used by resource manager servers. It is supported only on the Sun.

Classes

- **DsStTape**
- **DsStTapeConfig**

library DsStTapeShSh (shared)

/ecs/formal/DSS/stmgt/src/tp/tape

Classes

- **DsStTape**
- **DsStTapeConfig**

library EcCf

/ecs/formal/CSS/DOF/src/CF/cf

Classes

- **EcPfConfigFile** -- This class is used to read the Configuration file that holds the values of attributes needed for the process Framework.
- **ListAndNameCollect** -- This class is used to create collectable nodes whose fields are the attribute name and the linked list of the values for the respective attribute.

library EcCfSh (shared)

/ecs/formal/CSS/DOF/src/CF/cf

Classes

- **EcPfConfigFile** -- This class is used to read the Configuration file that holds the values of attributes needed for the process Framework.
- **ListAndNameCollect** -- This class is used to create collectable nodes whose fields are the attribute name and the linked list of the values for the respective attribute.

library EcClWbCgiVar

/ecs/formal/CLS/WKBCH/HtmlLib

This library contains utilities to read HTML templates and build HTML pages from the templates. It also provides classes that encapsulate the values that can be read into a CGI program.

Classes

- **ClWbCgiElement**
- **ClWbCgiVars**
- **ClWbTmplElement**
- **ClWbTmplVars**

library EcCsGateway

/ecs/formal/CSS/DOF/src/GATEWAY/src/common

Classes

- **CsGwGateway** -- This class may be used to send/receive gateway messages securely at a user specified level.
- **CsGwMessage** -- This class may be used to store and process the messages passing to and from Rel A communication gateway.
- **CsGwSdpfMsg** -- This derived class adds more functionalities to the CsGwMessage class as needed in processing messages for SDPF gateway.

library EcDcMsgPsng1

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/msg_oodce_1

Classes

- **EcMpFileInList** -- The EcMpFileInList object is the item stored in the persistent incoming message file.
- **EcMpFileOutList** -- The EcMpFileOutList object is the item stored in the persistent message file for incoming queue (callback or ordinary).
- **EcMpMsg** -- The EcMpMsg class contains the definitions for message object.
- **EcMpMsgCb** -- This class is an abstract ecs message callback class.
- **EcMpMsgCbList** -- Contains information about the callback object.
- **EcMpMsgIdList** -- Class containing the pair: msg id + receiver log name.
- **EcMpMsgPsngCtrl** -- The EcMpMsgPsngCtrl object is the controller object, through which any number of receiver sessions can be created.

- **EcMpMsgQueue** -- This class will be the parent class for the following.
- **EcMpMsgQueueCbIn** -- FileCleanup Clean the incoming callback persistent file and log in only those message currently in the callback queue EvalThread Get a request from the queue and process the request.
- **EcMpMsgQueueIn** -- This class will be used to queue the messages once they are received.
- **EcMpMsgQueueOut** -- In case of a asynchronous calls, the message is put into a queue, which will be sent later by an internal thread.
- **EcMpMsgWrapper** -- This class is used to wrap up the message and the related information that is common to all the receivers in a session list.
- **EcMpQueueInItemList** -- The EcMpQueueInItemList object is the item stored in the incoming queue (callback or ordinary queue).
- **EcMpQueueInList** -- Contains information about the incoming queues.
- **EcMpQueueOutItemList** -- The EcMpQueueOutItemList object is the item stored in the outgoing queue.
- **EcMpReceiverCacheList** -- Contains cached information about all the client pointers (sender list).
- **EcMpSessionItemList** -- List containing information about each receiver in the session list.
- **EcMpSessionList** -- This class allows to create a list of receivers to which message(s) will be sent.
- **EcMpTransfer_1_0** -- This definition declares the interface for this application and associates it with a globally (universally) unique identifier, or UUID.
- **EcMpTransfer_1_0_ABS** -- Abstract base class of EcMpTransfer.
- **EcMpTransfer_1_0_Mgr** -- Manager class for EcMpTransfer.
- **EcMpTransferCli** -- This class is the oodce EcMpTransfer client (surrogate object) It allows you to make requests to an EcMpTransfer manager object, and transfer messages.
- **EcMpTransferCliList** -- Contains information about the created receiver object.
- **EcMpTransferSrv** -- This class is the OODCE manager object.

library EcDcMsgPsng1Sh (shared)

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/msg_oodce_1

Classes

- **EcMpFileInList** -- The EcMpFileInList object is the item stored in the persistent incoming message file.
- **EcMpFileOutList** -- The EcMpFileOutList object is the item stored in the persistent message file for incoming queue (callback or ordinary).
- **EcMpMsg** -- The EcMpMsg class contains the definitions for message object.
- **EcMpMsgCb** -- This class is an abstract ecs message callback class.

- **EcMpMsgCbList** -- Contains information about the callback object.
- **EcMpMsgIdList** -- Class containing the pair: msg id + receiver log name.
- **EcMpMsgPsngCtrl** -- The EcMpMsgPsngCtrl object is the controller object, through which any number of receiver sessions can be created.
- **EcMpMsgQueue** -- This class will be the parent class for the following.
- **EcMpMsgQueueCbIn** -- FileCleanup Clean the incoming callback persistent file and log in only those message currently in the callback queue EvalThread Get a request from the queue and process the request.
- **EcMpMsgQueueIn** -- This class will be used to queue the messages once they are received.
- **EcMpMsgQueueOut** -- In case of a asynchronous calls, the message is put into a queue, which will be sent later by an internal thread.
- **EcMpMsgWrapper** -- This class is used to wrap up the message and the related information that is common to all the receivers in a session list.
- **EcMpQueueInItemList** -- The EcMpQueueInItemList object is the item stored in the incoming queue (callback or ordinary queue).
- **EcMpQueueInList** -- Contains information about the incoming queues.
- **EcMpQueueOutItemList** -- The EcMpQueueOutItemList object is the item stored in the outgoing queue.
- **EcMpReceiverCacheList** -- Contains cached information about all the client pointers (sender list).
- **EcMpSessionItemList** -- List containing information about each receiver in the session list.
- **EcMpSessionList** -- This class allows to create a list of receivers to which message(s) will be sent.
- **EcMpTransfer_1_0** -- This definition declares the interface for this application and associates it with a globally (universally) unique identifier, or UUID.
- **EcMpTransfer_1_0_ABS** -- Abstract base class of EcMpTransfer.
- **EcMpTransfer_1_0_Mgr** -- Manager class for EcMpTransfer.
- **EcMpTransferCli** -- This class is the oodce EcMpTransfer client (surrogate object) It allows you to make requests to an EcMpTransfer manager object, and transfer messages.
- **EcMpTransferCliList** -- Contains information about the created receiver object.
- **EcMpTransferSrv** -- This class is the OODCE manager object.

library EcDcNotify

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/notify/notify

Classes

- **EcCsNotify**
- **EcCsNotifyMsgCb** -- This class contains methods to handle the acknowledgment once a message is sent.

library EcDcNotifySh (shared)

/ecs/formal/CSS/DOF/src/MP_OODCE_NO1/notify/notify

Classes

- **EcCsNotify**
- **EcCsNotifyMsgCb** -- This class contains methods to handle the acknowledgment once a message is sent.

library EcDmDdClient

/ecs/formal/DM/DDICT/src/ddclient

The library establishes and maintains a session with the DDICT server.

Classes

- **DmDdClRequest** -- This is the signature of the callback function for asynchronous communication (via the Submit function).
- **DmDdClRequestServer** -- This class is used by a client to initiate a DDICT session.
- **DmDdClSchemaRequest** -- This is the signature of the callback function for asynchronous communication (via the Submit function).
- **DmDdClUrRequest** -- This is the signature of the callback function for asynchronous communication (via the Submit function).

library EcDmLmClient

/ecs/formal/DM/LIMGR/src/client

Classes

- **DmLmClRequest** -- Class contains all methods and attributes related to a user request.
- **DmLmClRequestServer** -- Class contains all methods and attributes related to tracking user's requests. User can:.

library EcDmLmLimServer

/ecs/formal/DM/LIMGR/src/server

The server CSC provides classes for DM subsystem servers, it contains classes for making LIM server. It makes the LIM server executable program EcDmLmServer. This library is server side library.

Classes

- **DmLmDispatcher** -- This is the customized version of the dispatcher object.
- **DmLmManagedServer** -- This class connects to the Process Framework infrastructure during initialization.
- **DmLmProcessPlanPair** -- This class is used by dispatcher to create DmLmProcessPlan vector list, the vector list in the DmLmDispatcher keeps the DmLmProcessPlan for the reuse.
- **DmLmSrRequest** -- This class contains all the methods related to a user request that is processed at the server side .

library EcDnDir

/ecs/formal/CSS/DOF/src/DN/directory

This provides classes to manipulate CDS entries, and operations on their attributes using XDS APIs.

Classes

- **CsDcXds** -- This class provides APIs to DCE-CDS application programming.
- **EcDnAttribute** -- This class represents an attribute of a CDS entry.
- **EcDnCompositeName** -- A composite name is a complete CDS path name.
- **EcDnContext** -- The context class defines the path/set of bindings with distinct atomic names.
- **EcDnElement** -- The EcDnElement class will contain an element, which is an attribute-value list pair.
- **EcDnValue** -- This class abstracts the value of an attribute.

library EcDnDirSh (shared)

/ecs/formal/CSS/DOF/src/DN/directory

This provides classes to manipulate CDS entries, and operations on their attributes using XDS APIs.

Classes

- **CsDcXds** -- This class provides APIs to DCE-CDS application programming.
- **EcDnAttribute** -- This class represents an attribute of a CDS entry.

- **EcDnCompositeName** -- A composite name is a complete CDS path name.
- **EcDnContext** -- The context class defines the path/set of bindings with distinct atomic names.
- **EcDnElement** -- The EcDnElement class will contain an element, which is an attribute-value list pair.
- **EcDnValue** -- This class abstracts the value of an attribute.

library EcDpJobMgmtClient

/ecs/formal/PDPS/DPS/PRONG/src/JobMgmtPF

The Job Management Client library provides the following public interfaces to the Job Management Server:

CreateDprJob - Creates a job directly in AutoSys, or on a queue if its data is not available.

ReleaseDprJob - Sets the state of a job to "Released" -- it is then eligible to be placed in AutoSys by the Job Management server if there is room.

CreateGEvntJob - Creates a ground event job directly in AutoSys.

CancelGEvntJob - Cancels a ground event job.

GetDprJobStatus - Gets the status of a job - started, success, failed, etc.

CancelDprJob - Cancels a job - either removes it from the queue or removes it from AutoSys if it is there, with database cleanup.

FreeAutoSys - Called whenever it is necessary to check the queue for jobs to put into Autosys -- this happens when a job is released, canceled, created or a job completes.

These library routines convert input into a message packet form suitable for communicating with the Job Management Server and then call OODCE routines to initiate and confirm the communication. The format of the messages is specified in the DpPrSchedulerDObj.idl file.

Classes

- **DpPrSchedulerDObj_1_0**
- **DpPrSchedulerProxy** -- Class.

library EcFh

/ecs/formal/CSS/DOF/src/FH/src

This library provides fault handling capabilities to ECS applications.

Classes

- **EcFhExecutor** -- This class invokes the EcFhFaultHandler::HandleDCEOSFException depending on the policy associated with this executor.
- **EcFhFaultHandler** -- This class knows how to handle exceptions of type DCEOSFException.
- **EcFhPolicy** -- This class creates the policy for fault handling.

library EcFhSh (shared)

/ecs/formal/CSS/DOF/src/FH/src

This library provides fault handling capabilities to ECS applications.

Classes

- **EcFhExecutor** -- This class invokes the EcFhFaultHandler::HandleDCEOSFException depending on the policy associated with this executor.
- **EcFhFaultHandler** -- This class knows how to handle exceptions of type DCEOSFException.
- **EcFhPolicy** -- This class creates the policy for fault handling.

library EcGwDAR

/ecs/formal/CSS/DOF/src/RELB_GATEWAY/DAR/dar

This library is the DAR Communications client library which client applications should link with to communicate with the DAR Gateway.

Classes

- **EcGwDARGetSubxARStatusRequest_C** -- This class represents the client side of DARGateway GetSubxARStatus.
- **EcGwDARGetxARStatusRequest_C** -- This class represents the client side of DARGateway GetxARStatus.
- **EcGwDARModifyDarRequest_C** -- This class represents the client side of DARGateway ModifyDar.
- **EcGwDARQueryxARContentsRequest_C** -- This class represents the client side of DARGateway QueryxARContents.
- **EcGwDARQueryxARScenesRequest_C** -- This class represents the client side of DARGateway QueryxARScenes.
- **EcGwDARQueryxARSummaryRequest_C** -- This class represents the client side of DARGateway QueryxARSummary.
- **EcGwDARSubmitDarRequest_C** -- This class represents the client side of DARGateway SubmitDar.
- **getSubxARStatusSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getSubxARStatus request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **getSubxARStatusSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getSubxARStatus result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .

- **getSubxARStatusSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getSubxARStatus status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **getxARStatusSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getxARStatus request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **getxARStatusSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getxARStatus result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **getxARStatusSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the getxARStatus status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **modifyDarSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the modifyDar request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **modifyDarSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the modifyDar result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **modifyDarSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the modifyDar status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **queryxARContentsSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARContents request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARContentsSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARContents result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARContentsSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARContents status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .

- **queryxARScenesSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARScenesSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARScenesSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARScenes status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARSummarySrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARSummarySrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **queryxARSummarySrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the queryxARSummary status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE .
- **submitDarSrfRequest** -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar request message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **submitDarSrfResult** -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar result message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.
- **submitDarSrfStatus** -- This message class, which is derived from the EcUtStreamable class, is used to transport the submitDar status message by the Server Request Framework (SRF), which is utilized by the ASTER DAR Gateway client library and server for DCE communication.

library EcHt

/ecs/formal/IOS/AdvService/src/EcHtmlLib

This library provides classes that contain operations to handle all HTML tags. Primarily, its functions are used to create dynamic HTML pages through CGI.

Classes

- **EcHtAddress** -- User interface to the EcHtAddressRep class.
- **EcHtAddressRep** -- Class which represents the HTML Address element.
- **EcHtBaseHandle** -- This class provides an abstract base class for a reference counting scheme.
- **EcHtCgiElement** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCgiElementKey** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCgiVars** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCode** -- User interface to the EcHtCodeRep class.
- **EcHtCodeRep** -- Class representing the HTML CODE element intended for short words or phrases of code; the PRE block structuring element is more appropriate for multiple-line listings.
- **EcHtCustomList** -- User interface to the EcHtCustomListRep class.
- **EcHtCustomListRep** -- This class provides an interface to the HTML unordered list construct.
- **EcHtDefinitionListRep** -- Class representing the HTML Defintion List (<DL>, <DT>, <DD>, and </DL>), a list of terms and corresponding defintions.
- **EcHtDirectoryList** -- User interface to the EcHtDirectoryListRep class.
- **EcHtDirectoryListRep** -- Class to represent the HTML Directory List (<OL>, <LI>, and </OL> elements).
- **EcHtElement** -- The handle class provides the client interface to the representation that holds the object's state. Object instances of this class should be declared normally, but operations should always be through the "operator->". This operator forwards operations
- **EcHtElementRep** -- This is the abstract base class for all HTML construct representation classes.
- **EcHtForm** -- User interface to the EcHtFormRep class.
- **EcHtFormCheckbox** -- User interface to the EcHtFormCheckboxRep class.
- **EcHtFormCheckboxRep** -- Class to represent a Form Check Box Input Element.
- **EcHtFormHidden** -- User interface to the EcHtFormHiddenRep class.
- **EcHtFormHiddenRep** -- Class to represent a Form Hidden Input Element.
- **EcHtFormImage** -- User interface to the EcHtFormImageRep class.
- **EcHtFormImageRep** -- Class to represent a Form Image Input Element.
- **EcHtFormPassword** -- User interface to the EcHtFormPasswordRep class.
- **EcHtFormPasswordRep** -- Class to represent a Form Passord Input Element.

- **EcHtFormRadio** -- User interface to the EcHtFormRadioRep class.
- **EcHtFormRadioRep** -- Class to represent a Form Radio Input Element.
- **EcHtFormRep** -- Class to represent an HTML Form.
- **EcHtFormReset** -- User interface to the EcHtFormResetRep class.
- **EcHtFormResetRep** -- Class to represent a Form Reset Input Element.
- **EcHtFormSelect** -- User interface to the EcHtFormSelectRep class.
- **EcHtFormSelectRep** -- Class to represent a Form Select Element.
- **EcHtFormSubmit** -- User interface to the EcHtFormSubmitRep class.
- **EcHtFormSubmitRep** -- Class to represent a Form Submit Input Element.
- **EcHtFormText** -- User interface to the EcHtFormTextRep class.
- **EcHtFormTextArea** -- Class to represent an HTML Form Text Area Element.
- **EcHtFormTextAreaRep** -- User interface to the EcHtFormTextAreaRep class.
- **EcHtFormTextRep** -- Class to represent an HTML Form Text Input Element.
- **EcHtHeader** -- User interface to the EcHtHeaderRep class.
- **EcHtHeaderRep** -- Class representing HTML <H>, </H> element for section headings.
- **EcHtImageRep** -- Class for an HTML IMG element for referring to an image or icon via a hyperlink.
- **EcHtLinkRep** -- Class to represent a hyperlink.
- **EcHtListRep** -- This is the abstract base class for all HTML lists.
- **EcHtOrderedList** -- User interface to the EcHtOrderedListRep class.
- **EcHtOrderedListRep** -- Class to represent the HTML Ordered List (<OL>, <LI>, and </OL> elements) for an ordered list of items (typically rendered as a numbered list).
- **EcHtPageRep** -- Class representing HTML page consisting of header, body and footer.
- **EcHtParagraph** -- User interface to the EcHtParagraphRep class.
- **EcHtParagraphRep** -- Class for representing an HTML <P> element for paragraphs.
- **EcHtPreformattedText --** User interface to the EcHtPreformattedText class.
- **EcHtPreformattedTextRep** -- Class representing the HTML Preformatted Text (<PRE> and </PRE> elements) for a character cell block of text and is suitable for text that has been formatted for a monospaced font.
- **EcHtRegion** -- User interface to the EcHtRegionRep class.
- **EcHtRegionRep** -- This class is a concrete base class to be used by callers/derived representations that contain many HTML elements.
- **EcHtTable** -- Provides the representation of the HTML Table tag and its attributes.
- **EcHtTableElements** -- Provides the representation of the HTML Table elements and their attributes.

- **EcHtUnorderedList** -- User interface to the EcHtUnorderedListRep class.
- **EcHtUnorderedListRep** -- Class representing the HTML Unordered List (<UL>, <LI>, and </UL> elements) for a list of items (typically rendered as a bulleted list).

library EcHtSh (shared)

/ecs/formal/IOS/AdvService/src/EcHtmlLib

This library provides classes that contain operations to handle all HTML tags. Primarily, its functions are used to create dynamic HTML pages through CGI.

Classes

- **EcHtAddress** -- User interface to the EcHtAddressRep class.
- **EcHtAddressRep** -- Class which represents the HTML Address element.
- **EcHtBaseHandle** -- This class provides an abstract base class for a reference counting scheme.
- **EcHtCgiElement** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCgiElementKey** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCgiVars** -- Class to parse arguments from a Web server for use by a cgi-bin program.
- **EcHtCode** -- User interface to the EcHtCodeRep class.
- **EcHtCodeRep** -- Class representing the HTML CODE element intended for short words or phrases of code; the PRE block structuring element is more appropriate for multiple-line listings.
- **EcHtCustomList** -- User interface to the EcHtCustomListRep class.
- **EcHtCustomListRep** -- This class provides an interface to the HTML unordered list construct.
- **EcHtDefinitionListRep** -- Class representing the HTML Defintion List (<DL>, <DT>, <DD>, and </DL>), a list of terms and corresponding defintions.
- **EcHtDirectoryList** -- User interface to the EcHtDirectoryListRep class.
- **EcHtDirectoryListRep** -- Class to represent the HTML Directory List (<OL>, <LI>, and </OL> elements).
- **EcHtElement** -- The handle class provides the client interface to the representation that holds the object's state. Object instances of this class should be declared normally, but operations should always be through the "operator->". This operator forwards operations
- **EcHtElementRep** -- This is the abstract base class for all HTML construct representation classes.
- **EcHtForm** -- User interface to the EcHtFormRep class.
- **EcHtFormCheckbox** -- User interface to the EcHtFormCheckboxRep class.

- **EcHtFormCheckboxRep** -- Class to represent a Form Check Box Input Element.
- **EcHtFormHidden** -- User interface to the EcHtFormHiddenRep class.
- **EcHtFormHiddenRep** -- Class to represent a Form Hidden Input Element.
- **EcHtFormImage** -- User interface to the EcHtFormImageRep class.
- **EcHtFormImageRep** -- Class to represent a Form Image Input Element.
- **EcHtFormPassword** -- User interface to the EcHtFormPasswordRep class.
- **EcHtFormPasswordRep** -- Class to represent a Form Passord Input Element.
- **EcHtFormRadio** -- User interface to the EcHtFormRadioRep class.
- **EcHtFormRadioRep** -- Class to represent a Form Radio Input Element.
- **EcHtFormRep** -- Class to represent an HTML Form.
- **EcHtFormReset** -- User interface to the EcHtFormResetRep class.
- **EcHtFormResetRep** -- Class to represent a Form Reset Input Element.
- **EcHtFormSelect** -- User interface to the EcHtFormSelectRep class.
- **EcHtFormSelectRep** -- Class to represent a Form Select Element.
- **EcHtFormSubmit** -- User interface to the EcHtFormSubmitRep class.
- **EcHtFormSubmitRep** -- Class to represent a Form Submit Input Element.
- **EcHtFormText** -- User interface to the EcHtFormTextRep class.
- **EcHtFormTextArea** -- Class to represent an HTML Form Text Area Element.
- **EcHtFormTextAreaRep** -- User interface to the EcHtFormTextAreaRep class.
- **EcHtFormTextRep** -- Class to represent an HTML Form Text Input Element.
- **EcHtHeader** -- User interface to the EcHtHeaderRep class.
- **EcHtHeaderRep** -- Class representing HTML <H>, </H> element for section headings.
- **EcHtImageRep** -- Class for an HTML IMG element for referring to an image or icon via a hyperlink.
- **EcHtLinkRep** -- Class to represent a hyperlink.
- **EcHtListRep** -- This is the abstract base class for all HTML lists.
- **EcHtOrderedList** -- User interface to the EcHtOrderedListRep class.
- **EcHtOrderedListRep** -- Class to represent the HTML Ordered List (<OL>, <LI>, and </OL> elements) for an ordered list of items (typically rendered as a numbered list).
- **EcHtPageRep** -- Class representing HTML page consisting of header, body and footer.
- **EcHtParagraph** -- User interface to the EcHtParagraphRep class.
- **EcHtParagraphRep** -- Class for representing an HTML <P> element for paragraphs.
- **EcHtPreformattedText** -- User interface to the EcHtPreformattedText class.

- **EcHtPreformattedTextRep** -- Class representing the HTML Preformatted Text (<PRE> and </PRE> elements) for a character cell block of text and is suitable for text that has been formatted for a monospaced font.
- **EcHtRegion** -- User interface to the EcHtRegionRep class.
- **EcHtRegionRep** -- This class is a concrete base class to be used by callers/derived representations that contain many HTML elements.
- **EcHtTable** -- Provides the representation of the HTML Table tag and its attributes.
- **EcHtTableElements** -- Provides the representation of the HTML Table elements and their attributes.
- **EcHtUnorderedList** -- User interface to the EcHtUnorderedListRep class.
- **EcHtUnorderedListRep** -- Class representing the HTML Unordered List (<UL>, <LI>, and </UL> elements) for a list of items (typically rendered as a bulleted list).

library EcNsServiceLoc

/ecs/formal/CSS/DOF/src/NS/naming

This library contains both the client and server side APIs to allow servers to advertise their services and to allow clients to locate the servers. The primary server side functions are ExportBinding and GroupMemberAdd. These functions allow a server to advertise its services in CDS. The primary client side function is GetServerBinding which is responsible for querying CDS for a desired service and returning a valid binding to the application.

Classes

- **EcNsServiceLoc** -- This class captures the common operations and state used by the client and server derivatives of this class.
- **EcNsServiceLocClient** -- The class supports multiple constructors to accomodate various means of specifying a service.
- **EcNsServiceLocServer** -- This class constructs a CDS path for a server, allows that path to be added to an RPC group and allows both the RPC group and the server path to exported to CDS.

library EcNsServiceLocator

/ecs/formal/CSS/DOF/src/NS/service_locator

The Service Locator library provides an implementation independent mechanism for specifying service locations. The mechanism is called the service token and essentially consists of a cell name and a service name. There are two parts to the library. The first is the service locator class which encapsulates the service token management APIs. The second is the locator mapper class which provides access to a service locator mapping file. The mapping file maps service/cell long names to shortened versions of the name. For instance, the cell /.../gsfccell.gsfcb.nasa.gov is mapped to GSF and the service name EcSbSubServer is mapped to CSSBSRV.

Classes

- **EcNsLocatorMapper** -- This class maps long CDS entries like cell names and server group entries to and from short representations.
- **EcNsServiceLocator** -- This Class provide technology independant means of describing an ECS service.

library EcNsServiceLocatorSh (shared)

/ecs/formal/CSS/DOF/src/NS/service_locator

The Service Locator library provides an implementation independent mechanism for specifying service locations. The mechanism is called the service token and essentially consists of a cell name and a service name. There are two parts to the library. The first is the service locator class which encapsulates the service token management APIs. The second is the locator mapper class which provides access to a service locator mapping file. The mapping file maps service/cell long names to shortened versions of the name. For instance, the cell /.../gsfccell.gsfcb.nasa.gov is mapped to GSF and the service name EcSbSubServer is mapped to CSSBSRV.

Classes

- **EcNsLocatorMapper** -- This class maps long CDS entries like cell names and server group entries to and from short representations.
- **EcNsServiceLocator** -- This Class provide technology independant means of describing an ECS service.

library EcNsServiceLocSh (shared)

/ecs/formal/CSS/DOF/src/NS/naming

This library contains both the client and server side APIs to allow servers to advertise their services and to allow clients to locate the servers. The primary server side functions are ExportBinding and GroupMemberAdd. These functions allow a server to advertise its services in CDS. The primary client side function is GetServerBinding which is responsible for querying CDS for a desired service and returning a valid binding to the application.

Classes

- **EcNsServiceLoc** -- This class captures the common operations and state used by the client and server derivatives of this class.
- **EcNsServiceLocClient** -- The class supports multiple constructors to accomodate various means of specifying a service.
- **EcNsServiceLocServer** -- This class constructs a CDS path for a server, allows that path to be added to an RPC group and allows both the RPC group and the server path to exported to CDS.

library EcPf

/ecs/formal/CSS/DOF/src/PF/pf

Classes

- **EcPfClient**
- **EcPfGenProcess** -- EcPfGenProcess is an abstract class which represents the process framework for a generic process.
- **EcPfGenServer** -- The class will provide the generic DCEServer capabilities and some enhanced features as well.
- **EcPfManagedServer**
- **EcPfProxy** -- Currently does not maintain any information or provide any functionality.
- **EcPfUnmanagedServer** -- Pointer to a PfClient, In otherwords it is a pointer to itself.
- **ObjectLinkList** -- This class maintains a linked list of structures ObjectLink whose fields are DCEObject and a pointer to the next structure in the list.

library EcPfSh (shared)

/ecs/formal/CSS/DOF/src/PF/pf

Classes

- **EcPfClient**
- **EcPfGenProcess** -- EcPfGenProcess is an abstract class which represents the process framework for a generic process.
- **EcPfGenServer** -- The class will provide the generic DCEServer capabilities and some enhanced features as well.
- **EcPfManagedServer**
- **EcPfProxy** -- Currently does not maintain any information or provide any functionality.
- **EcPfUnmanagedServer** -- Pointer to a PfClient, In otherwords it is a pointer to itself.
- **ObjectLinkList** -- This class maintains a linked list of structures ObjectLink whose fields are DCEObject and a pointer to the next structure in the list.

library EcPo

/ecs/formal/IOS/AdvService/src/PersistentLib

The Persistent CSC provides a set of abstract and base classes for approved advertisments and requests. It also provides classes which support caching and defer fetching mechanism.

Classes

- **EcPoCache** -- This class supports a cache of different different subclasses of EcPoIndentifiedRepresentation.
- **EcPoCachingHandle** -- This class works in concert with EcPoIdentifiedRepresentation to support a cache of representation in addition to acting as a EcPoIdentifiedHandle.

- **EcPoCoordFetch** -- This class is the abstract base class for all objects that are RWDB Tools to do their persistence and allow other objects to coordinate thier fetches with it to allow for multiple objects to be fetched form the database with a single hit.
- **EcPoDependant** -- If a object contains another object it is typical for the life of the sub-object to be constrained by the super-object.
- **EcPoHandle** -- This class works in concert with EcPoRepresentation as abstract base classes for a persistent object framework.
- **EcPoIdentifiedHandle** -- This class works in concert with EcPoIdenitifiedRepresentation to provide a convenient deferred fetching handle with an integer ID.
- **EcPoIdentifiedRepresentation** -- We are a specialization of a EcPoRepresentation with the concept of a defined primary key, our IdentityState.
- **EcPoIdentityState** -- Private View.
- **EcPoLongString** -- This class is a RWCString that can insert itself into the database.
- **EcPoPersistent** -- Abstract base class for all persistent objects.
- **EcPoRepresentation** -- This class is the abstract base class for all objects that are persistent and are part of the deferred fetching scheme.
- **EcPoUpdateAware** -- This class is a specialization of the EcPoCoordFetch class which adds the capability of being aware of the UPDATE capability of the database by inheriting an ID.

library EcPoDbRW

/ecs/formal/COMMON/CSCI_DBWrapper/src

Classes

- **EcPoConnectionsRW** -- This class provides access to specific global objects by providing the handle for them.
- **EcPoDb** -- Objects of this class represent an RWDB Database together with the connections in use by the database.
- **EcPoError** -- This class provides a series of misc.
- **EcPoStrategy** -- Objects of this class represent error recovery strategies.
- **EcPoThread** -- Objects of this class represent a single RWDB connection and the thread ID of the thread that claimed the connection.

library EcPoDbRWSh (shared)

/ecs/formal/COMMON/CSCI_DBWrapper/src

Classes

- **EcPoConnectionsRW** -- This class provides access to specific global objects by providing the handle for them.

- **EcPoDb** -- Objects of this class represent an RWDB Database together with the connections in use by the database.
- **EcPoError** -- This class provides a series of misc.
- **EcPoStrategy** -- Objects of this class represent error recovery strategies.
- **EcPoThread** -- Objects of this class represent a single RWDB connection and the thread ID of the thread that claimed the connection.

library EcPoSh (shared)

/ecs/formal/IOS/AdvService/src/PersistentLib

The Persistent CSC provides a set of abstract and base classes for approved advertisments and requests. It also provides classes which support caching and defer fetching mechanism.

Classes

- **EcPoCache** -- This class supports a cache of different different subclasses of EcPoIndentifiedRepresentation.
- **EcPoCachingHandle** -- This class works in concert with EcPoIdentifiedRepresentation to support a cache of representation in addition to acting as a EcPoIdentifiedHandle.
- **EcPoCoordFetch** -- This class is the abstract base class for all objects that are RWDB Tools to do their persistence and allow other objects to coordinate thier fetches with it to allow for multiple objects to be fetched form the database with a single hit.
- **EcPoDependant** -- If a object contains another object it is typical for the life of the sub-object to be constrained by the super-object.
- **EcPoHandle** -- This class works in concert with EcPoRepresentation as abstract base classes for a persistent object framework.
- **EcPoIdentifiedHandle** -- This class works in concert with EcPoIdenitifiedRepresentation to provide a convenient deferred fetching handle with an integer ID.
- **EcPoIdentifiedRepresentation** -- We are a specialization of a EcPoRepresentation with the concept of a defined primary key, our IdentityState.
- **EcPoIdentityState** -- Private View.
- **EcPoLongString** -- This class is a RWCString that can insert itself into the database.
- **EcPoPersistent** -- Abstract base class for all persistent objects.
- **EcPoRepresentation** -- This class is the abstract base class for all objects that are persistent and are part of the deferred fetching scheme.
- **EcPoUpdateAware** -- This class is a specialization of the EcPoCoordFetch class which adds the capability of being aware of the UPDATE capability of the database by inheriting an ID.

library EcPtSpecialLock

/ecs/formal/CSS/DOF/src/THREAD/locks

This library provides classes for mutex operations and thread IDs.

Classes

- **EcPtSharedLock** -- EcPtSharedLock is a class of shared/exclusive mutexes.
- **EcPtThreadId** -- This is an ID for a thread.
- **EcPtUnlockingMutex** -- EcPtUnlockingMutex is a class of self unlocking mutexes.

library EcPtSpecialLockSh (shared)

/ecs/formal/CSS/DOF/src/THREAD/locks

This library provides classes for mutex operations and thread IDs.

Classes

- **EcPtSharedLock** -- EcPtSharedLock is a class of shared/exclusive mutexes.
- **EcPtThreadId** -- This is an ID for a thread.
- **EcPtUnlockingMutex** -- EcPtUnlockingMutex is a class of self unlocking mutexes.

library ECSAuthn

/ecs/formal/COMMON/CSCI_Util/src/EcsAuthenticator

library EcSbCl

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/cl

This library contains all the classes for Subscription Server client.

Classes

- **EcClCancelAllSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClCancelEventIDSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClCancelExpDateSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClCancelSubCb**
- **EcClCancelUserIDSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClEvent**
- **EcClEventCollector** -- This class provides a collection mechanism for retrieving and manipulating multiple events.
- **EcClGetAllEventsCb** -- Class for obtaining subscription data.
- **EcClGetAllSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClGetCatEventsCb** -- Class for obtaining subscription data.
- **EcClGetCreatorEventsCb** -- Class for obtaining subscription data.
- **EcClGetEventDataCb**
- **EcClGetEventIDSubscriptionsCb** -- Class for obtaining subscription data.

- **EcClGetExpDateSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClGetSubDataCb** -- Class for obtaining subscription data.
- **EcClGetUserIDSubscriptionsCb** -- Class for obtaining subscription data.
- **EcClRegisterEventCb**
- **EcClRestartNotifyCb**
- **EcClSubmitSubCb**
- **EcClSubscription** -- Client side subscription.
- **EcClSubscriptionCollector** -- This class provides a collection mechanism for retrieving and manipulating multiple subscriptions.
- **EcClTriggerEventCb**
- **EcClUnregisterAllEventsCb** -- Class for obtaining subscription data.
- **EcClUnregisterCatEventsCb** -- Class for obtaining subscription data.
- **EcClUnregisterCreatorEventsCb** -- Class for obtaining subscription data.
- **EcClUnregisterEventCb**
- **EcClUpdateEventCb**
- **EcClUpdateSubCb**

library EcSbMsg

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/msg

Classes

- **EcSbAcquireAction** -- This is a subclass of EcSbGenAction which is derived from EcUtStreamable.
- **EcSbEventUR** -- The EcSbEventUR class is derived from EcUrUR to provide an EventUR type class for Event server to use.
- **EcSbEventURProvider** -- Data entity enables this class to quickly retrieve the object parameters.
- **EcSbGenAction** -- All the actions specified by subscriber should subclass from it.
- **EcSbNotification** -- This class embeds all notification information which server uses to notify user when action is taken place.
- **EcSbSubUR**
- **EcSbSubURProvider**
- **MsgCancelAllSubscriptions**
- **MsgCancelEvent**
- **MsgCancelEventIDSubscriptions**
- **MsgCancelExpDateSubscriptions**

- **MsgCancelSubscription**
- **MsgCancelUserIDSubscriptions**
- **MsgGenResult**
- **MsgGenStatus**
- **MsgGetAllEvents**
- **MsgGetAllSubscriptions**
- **MsgGetCatEvents**
- **MsgGetCreatorEvents**
- **MsgGetEventData**
- **MsgGetEventDataResult**
- **MsgGetEventIDSubscriptions**
- **MsgGetExpDateSubscriptions**
- **MsgGetSubDataResult**
- **MsgGetSubscriptionData**
- **MsgGetUserIDSubscriptions**
- **MsgListResult**
- **MsgRegisterEvent**
- **MsgRegisterEventResult**
- **MsgRestartNotify**
- **MsgSubmitSubResult**
- **MsgSubmitSubscription**
- **MsgTriggerEvent**
- **MsgUnregisterAllEvents**
- **MsgUnregisterCatEvents**
- **MsgUnregisterCreatorEvents**
- **MsgUpdateEvent**
- **MsgUpdateSubscription**

library EcSbSr

/ecs/formal/CSS/DOF/src/SUBSCRIPTION/sb

This library consists of all subscription server side class objects. It is used by EcSbSubServer program only.

Classes

- **EcSbCancelAllSubsRequest** -- Handles a request of cancelling all subscription by the server.
- **EcSbCancelEventIDSubsRequest** -- Handles a request of cancelling all subscription associated with a given event by the server.
- **EcSbCancelEventRequest** -- Handles a request of cancelling a specified event by the server.
- **EcSbCancelExpDateSubsRequest** -- Handles a request of cancelling all subscription whose expiration date is beyond specified date.
- **EcSbCancelSubRequest** -- Handles a request of cancelling a specified subscription by the server.
- **EcSbCancelUserIDSubsRequest** -- Handles a request of cancelling all subscription subscribed by a specified user.
- **EcSbEvent** -- This is the server side event representation, customized to implement the actual operations supported by the server.
- **EcSbEventDispatcher** -- This is the customized version of the dispatcher object.
- **EcSbEventHandler** -- This class provides the interface to work with multiple events (i.e., aggregate behaviour) on the server side.
- **EcSbEventRServer** -- This is a wrapper class to wrape up EcSrRequestServer_S for the Event Server Object.
- **EcSbEventStore** -- This class provides the means for adding events to the persistent store, instantiating events from the persistent store, and passing the events to the subscription server via EcSbEventHandler.
- **EcSbGetAllEventsRequest** -- Handles a request of getting all events by the server.
- **EcSbGetAllSubsRequest** -- Handles a request of getting all subscription by the server.
- **EcSbGetCatEventsRequest** -- Handles a request of getting an event by given event category.
- **EcSbGetCreatorEventsRequest** -- Handles a request of getting all events created by a given user.
- **EcSbGetEventIDSubsRequest** -- Handles a request of getting all subscription associated with given event id.
- **EcSbGetEventRequest** -- Handles a request of getting all events from server.

- **EcSbGetExpDateSubsRequest** -- Handles a request of getting all subscription by given expiration data.
- **EcSbGetSubRequest** -- Handles a request of getting all subscription by the server.
- **EcSbGetUserIDSubsRequest** -- Handles a request of retrieve subscription subscribed by a given user.
- **EcSbManagedServer** -- EcSbManagedServer is a class which is dervived from EcPfManagedServer.
- **EcSbRegisterEventRequest** -- This class represent an event register request in subscription server.
- **EcSbRestartNotifyRequest**
- **EcSbSubmitSubRequest** -- Handles a request of submitting a subscription a client.
- **EcSbSubscription** -- This is the server side subscription representation.
- **EcSbSubscriptionDispatcher** -- This is the customized version of the dispatcher object.
- **EcSbSubscriptionHandler** -- This class provides the mechanism for collecting and passing around EcSbSubscriptions.
- **EcSbSubscriptionRServer** -- This is the server wrapper class.
- **EcSbSubscriptionStore** -- This class provides the means for adding subscriptions to the persistent store, instantiating subscriptions from the persistent store, and passing the subscriptions to the subscription server via EcSbSubscriptionHandler.
- **EcSbTimeKeeper** -- This class keeps track of time as the subscription server runs and makes sure that anything linked to time (such as the expiration of subscriptions) occurs.
- **EcSbTriggerEventRequest** -- Handles a request of triggering an event by the server.
- **EcSbUnregisterAllEventsRequest** -- Handles a request of cancelling all events by the server.
- **EcSbUnregisterCatEventsRequest** -- Handles a request of cancelling all events related to specified category.
- **EcSbUnregisterCreatorEventsRequest** -- Handlea a request of cancelling all event related to a specified user.
- **EcSbUpdateEventRequest** -- Handles a request of updating an event by the server.
- **EcSbUpdateSubRequest** -- Handles a request of updating a subscription by server.

library EcSeAuthn

/ecs/formal/CSS/DOF/src/SEC/authentication

The security authentication library allows a server to obtain a DCE login key via user input. Normally servers obtain their key by reading it from a keytab file. But in the event that a server should require manual login to DCE, this library provides that functionality. This class works with the EcSeLoginContext class to create a DCE login context for a process.

Classes

- **EcSeAuthn** -- The EcSeAuthn class inherits from the DCEMemPassword class which is the default implementation of the DCEPassword base class.

library EcSeAuthnSh (shared)

/ecs/formal/CSS/DOF/src/SEC/authentication

The security authentication library allows a server to obtain a DCE login key via user input. Normally servers obtain their key by reading it from a keytab file. But in the event that a server should require manual login to DCE, this library provides that functionality. This class works with the EcSeLoginContext class to create a DCE login context for a process.

Classes

- **EcSeAuthn** -- The EcSeAuthn class inherits from the DCEMemPassword class which is the default implementation of the DCEPassword base class.

library EcSeCmi

/ecs/formal/CSS/DOF/src/AUTHN/authn

Cryptographic Management Interface (CMI) provides a means for processes to obtain randomized passwords with which to gain access to non-DCE services (e.g., Sybase). This library generates a randomized username and password based on a given seed. The seed is combined with a data file of highly ramdomized data (EcSeRandomDataFile) and uses an indexing algorithm to obtain a randomized username and password.

Classes

- **EcSeCmi** -- This class will generate a user ID and password based on a given application key.

library EcSeCmiSh (shared)

/ecs/formal/CSS/DOF/src/AUTHN/authn

Cryptographic Management Interface (CMI) provides a means for processes to obtain randomized passwords with which to gain access to non-DCE services (e.g., Sybase). This library generates a randomized username and password based on a given seed. The seed is combined with a data file of highly ramdomized data (EcSeRandomDataFile) and uses an indexing algorithm to obtain a randomized username and password.

Classes

- **EcSeCmi** -- This class will generate a user ID and password based on a given application key.

library EcSeLogin

/ecs/formal/CSS/DOF/src/SEC/login/loginlib

This library creates a DCE login context for a process. It differs from the EcSeAuthn and EcSeServerKeyMgmt libraries in that it creates the login context by fork/exec'ing the dce_login process. This library is only primarily used by the EcSeLoginProg DCE login utility.

library EcSeLoginContext

/ecs/formal/CSS/DOF/src/SEC/logincontext

This library uses the key obtained from either the EcSeServerKeyMgmt or EcSeAuthn library to establish a login context for a process.

Classes

- **EcSeLoginContext** -- Any file that uses EcSeLoginContext must include this header file.

library EcSeLoginContextSh (shared)

/ecs/formal/CSS/DOF/src/SEC/logincontext

This library uses the key obtained from either the EcSeServerKeyMgmt or EcSeAuthn library to establish a login context for a process.

Classes

- **EcSeLoginContext** -- Any file that uses EcSeLoginContext must include this header file.

library EcSeSecurity

/ecs/formal/CSS/DOF/src/SEC/security

This library is required to to implement security service in any application.

Classes

- **AclInfo**
- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **EcSeDBAcl** -- This is the header file which defines the EcSeDBAcl class.
- **EcSeDBAclEntry** -- The EcSeDBAclEntry class represents the information about an ACL.
- **EcSeDBApp** -- The EcSeDBApp class serves as an encapsulation of a table on a database containing application to ACL DB correspondence information matching the format of a EcSeDBAppEntry instance.
- **EcSeDBAppEntry** -- The EcSeDBAppEntry class represents the relationship between an application and an ACL DB.

- **EcSeDBEntry** -- The EcSeDBEntry class serves as an encapsulation of a table on a database containing ACL entry information matching the format of a EcSeDBEntryEntry instance.
- **EcSeDBEntryEntry** -- The EcSeDBEntryEntry class represents an ACL entry.
- **EcSeDBSchema** -- The EcSeDBSchema class serves as an encapsulation of a table on a database containing ACL DB and schema information matching the format of a EcSeDBSchemaEntry instance.
- **EcSeDBSchemaEntry** -- The EcSeDBSchemaEntry class represents the relationship between an ACL DB and its schema.
- **EcSeGenSecurity** -- Any file that use the EcSeGenSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **EcSePersistAclStorage** -- EcSePersistAclStorage implements the persistent storage of ACL using Sybase.
- **EcSeSecurity** -- Any file that use the EcSeSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **SchemaInfo**

library EcSeServerKeyMgmt

/ecs/formal/CSS/DOF/src/SEC/keymgmt

This library provides a mechanism for servers to read their DCE login key from a keytab file. This is the most used method by which servers login to DCE. Keytab files are stored in the $ECS_HOME/CUSTOM/security directory.

Classes

- **EcSeServerKeyMgmt** -- The EcSeServerKeyMgmt class is the concrete implementation of the abstract DCEPassword class.

library EcSeServerKeyMgmtSh (shared)

/ecs/formal/CSS/DOF/src/SEC/keymgmt

This library provides a mechanism for servers to read their DCE login key from a keytab file. This is the most used method by which servers login to DCE. Keytab files are stored in the $ECS_HOME/CUSTOM/security directory.

Classes

- **EcSeServerKeyMgmt** -- The EcSeServerKeyMgmt class is the concrete implementation of the abstract DCEPassword class.

library EcSeSybSecurityStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **AclInfo**
- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **EcSeGenSecurity** -- Any file that use the EcSeGenSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **EcSeSybSecurity** -- Any file that use the EcSeSybSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **SchemaInfo**

library EcSeSybSecurityStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **AclInfo**
- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **EcSeGenSecurity** -- Any file that use the EcSeGenSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **EcSeSybSecurity** -- Any file that use the EcSeSybSecurity class, its methods or its global variable EcXSeSecurity must include this header file.
- **SchemaInfo**

library EcSeUtilityCtStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **EcDbInterface** -- The following functions allow users to connect to database, execute SQL statements, verify connections states, and disconnect from the database.
- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **FetchAclInfo**

library EcSeUtilityCtStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **EcDbInterface** -- The following functions allow users to connect to database, execute SQL statements, verify connections states, and disconnect from the database.

- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **FetchAclInfo**

library EcSeUtilityDbStubbed

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **FetchAclInfo**

library EcSeUtilityDbStubbedSh (shared)

/ecs/formal/CSS/DOF/src/SEC/sybsecuritystubbs

Classes

- **ECS_sec_id_pac_t** -- This class contains methods to write and read the sec_id_pac_t into and from a memory stream.
- **FetchAclInfo**

library EcTiTime

/ecs/formal/CSS/DOF/src/TIME/time

This library is used to obtain the current time in various formats, and manipulate the time. There will be no capability for the programmer to set the time.

Classes

- **EcTiTimeService** -- This class is used to obtain the current time in various formats, and manipulate the time.

library EcTiTimeSh (shared)

/ecs/formal/CSS/DOF/src/TIME/time

This library is used to obtain the current time in various formats, and manipulate the time. There will be no capability for the programmer to set the time.

Classes

- **EcTiTimeService** -- This class is used to obtain the current time in various formats, and manipulate the time.

library EcUr

/ecs/formal/COMMON/CSCI_UR/src/UR/framework

A Universal Reference (UR) is created as a logical reference to a extant ECS object. This reference can be passed around, written to disk, read from disk, etc. A UR is used when processing it is considered easier/cheaper than processing the actual object. Later the actual object can be reconstituted from the UR. URs are used to represent at LEAST the following in ECS:

- Data Granules
- Advertisements
- Sessions

Application programs that interact with the user will provide URs to users in lieu of the objects the application processes. These URs can be stored in files, e-mailed, etc.

Detailed design documentation can be found in the Release B 305 Overview documentation. Additional material is available from the current maintainer.

These classes are fully thread-safe. Accesses to static data are protected by mutexes.

Classes

- **EcUrServerUR** -- The EcUrServerUR class, which is inherited from the EcUrUR class, holds additional attributes and overrides EcUrUR methods in order to support Server URs.
- **EcUrServerURProvider** -- The EcUrServerURProvider class, which is inherited from the EcUrURProvider class, holds additional attributes and overrides EcUrURProvider methods in order to support Server UR providers.
- **EcUrUR** -- The Universal Reference (UR) is the abstract base class for all references to UR Providers.
- **EcUrURMaker** -- This class has the responsibility to aid working with objects that subclass UR that are either abstract or whose type is not known.
- **EcUrURProvider** -- This class is the abstract base class for all things refered to by Universal Reference (UR)s.
- **EcUrURProviderMaker** -- This class supports the creation of fully reconstituted UR providers given a UR.

library EcUrSh (shared)

/ecs/formal/COMMON/CSCI_UR/src/UR/framework

A Universal Reference (UR) is created as a logical reference to a extant ECS object. This reference can be passed around, written to disk, read from disk, etc. A UR is used when processing it is considered easier/cheaper than processing the actual object. Later the actual object can be reconstituted from the UR. URs are used to represent at LEAST the following in ECS:

- Data Granules
- Advertisements
- Sessions

Application programs that interact with the user will provide URs to users in lieu of the objects the application processes. These URs can be stored in files, e-mailed, etc.

Detailed design documentation can be found in the Release B 305 Overview documentation. Additional material is available from the current maintainer.

These classes are fully thread-safe. Accesses to static data are protected by mutexes.

Classes

- **EcUrServerUR** -- The EcUrServerUR class, which is inherited from the EcUrUR class, holds additional attributes and overrides EcUrUR methods in order to support Server URs.
- **EcUrServerURProvider** -- The EcUrServerURProvider class, which is inherited from the EcUrURProvider class, holds additional attributes and overrides EcUrURProvider methods in order to support Server UR providers.
- **EcUrUR** -- The Universal Reference (UR) is the abstract base class for all references to UR Providers.
- **EcUrURMaker** -- This class has the responsibility to aid working with objects that subclass UR that are either abstract or whose type is not known.
- **EcUrURProvider** -- This class is the abstract base class for all things refered to by Universal Reference (UR)s.
- **EcUrURProviderMaker** -- This class supports the creation of fully reconstituted UR providers given a UR.

library EcUt

/ecs/formal/COMMON/CSCI_Util/src/Logging

Classes

- **EcLgErrorEvent** -- This object provides a simple mechanism for converting a light-weight uuid_s_ok error message object (EcLgErrorMsg) into an EcAgEvent.
- **EcLgErrorMsg** -- This object encapsulates an ECS error event in a more lightweight contain than the EcAgEvent.
- **EcUtErrorDetails** -- EcUtErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- **EcUtRpcID** -- EcUtRpcID is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- **EcUtStreamable** -- The EcUtStreamable class provides an reuseable interface for ÿ// objects which need to be transferred between processes.
- **EcUtUniqueID**

library EcUtFactory

/ecs/formal/COMMON/CSCI_Util/src/Factory

The **EcUtFactory** should be used when you dynamically create objects based on the a class identifier.

EcUtFactory is a template class that allows you to specify the base class to use as the root of all objects created. In addition, #define macros are provided that allow derived classes to automatically be "register" with the factory with no additional programming.

The **EcUtFactory** support operations to Register a new class and Make an object of that class.

For those familiar with RogueWave collectibles, this class offers similar capabilities but is lighter-weight because it does not require defining save-guts/restore-guts. It is more flexible in several ways. It is template based allowing for an arbitrary base class. It allows for strings or integers as the classID. It support dynamic registration in a multi-threaded environment (GuardedFactory only).

Classes

- **EcUtClassID** -- This object allows an class identifier to be created with a value, this identifier to be assigned, compared, and checks if it is valid.

Templates

- **EcUtFactory** -- This class supports the registering and execution of functions to create subclasses of <BaseClass>.
- **EcUtGuardedFactory** -- This class supports the creation of subclasses of <BaseClass> in a threaded environment.

library EcUtFactorySh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Factory

The **EcUtFactory** should be used when you dynamically create objects based on the a class identifier.

EcUtFactory is a template class that allows you to specify the base class to use as the root of all objects created. In addition, #define macros are provided that allow derived classes to automatically be "register" with the factory with no additional programming.

The **EcUtFactory** support operations to Register a new class and Make an object of that class.

For those familiar with RogueWave collectibles, this class offers similar capabilities but is lighter-weight because it does not require defining save-guts/restore-guts. It is more flexible in several ways. It is template based allowing for an arbitrary base class. It allows for strings or integers as the classID. It support dynamic registration in a multi-threaded environment (GuardedFactory only).

The class **EcUtGuardedFactory** is a fully thread-safe version of **EcUtFactory**. Accesses to the factory are protected by mutexes.

Classes

- **EcUtClassID --** This object allows an class identifier to be created with a value, this identifier to be assigned, compared, and checks if it is valid.

Templates

- **EcUtFactory** -- This class supports the registering and execution of functions to create subclasses of <BaseClass>.
- **EcUtGuardedFactory** -- This class supports the creation of subclasses of <BaseClass> in a threaded environment.

library EcUtGUIg_mt_eh_tp

/ecs/formal/COMMON/CSCI_GUI/src

Classes

- **EcRwCollectable** -- The EcRwCollectable class is an absract base class derived from Rogue Wave's collectable class. It does not redefine any of the inherited functionality but adds IsInheritedFrom method.
- **EcUtDateEntry**
- **EcUtDateInterval**
- **EcUtDblSelBox**
- **EcUtEpakTable**
- **EcUtIntegerTextField**
- **EcUtListbox**
- **EcUtRangeCheck** -- EcUtRangeCheck class is used to check that a value is within a certain range.
- **EcUtRealTextField**
- **EcUtSelectableItem** -- It contains an identifying string and points to a contained object.
- **EcUtSelectableItemContainer** -- This class is intended to store a list of EcUtSelectableItems.
- **EcUtSelectableItemState** -- Declares the EcUtSelectableItemState class.
- **EcUtSetTabChildSaved**
- **EcUtSetTextWidgetEditable**
- **EcUtTextField**
- **EcUtTimeOnlyEntry**
- **EcUtTimeOnlyInterval**
- **TimeEntry**
- **TimeInterval**

library EcUtHelpg_eh_tp

/ecs/formal/COMMON/CSCI_HELP/src/HelpClass

Classes

- **EcUtCIHelp**

library EcUtHelpg_mt_eh_tp

/ecs/formal/COMMON/CSCI_HELP/src/HelpClass

Classes

- **EcUtCIHelp**

library EcUtMisc

/ecs/formal/COMMON/CSCI_Util/src/Misc

This library contains miscellaneous utilities.

Classes

- **EcUtFileCopy** -- Objects of this class copy a single file from the specified source location to a specified destination location.
- **EcUtFileManager** -- This class supports expanding special characters in filenames.
- **EcUtPerfData** -- Log Performance Data.

library EcUtMiscSh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Misc

The EcUtMisc library contains miscellaneous utilities.

Classes

- **EcUtFileCopy** -- Objects of this class copy a single file from the specified source location to a specified destination location.
- **EcUtFileManager** -- This class supports expanding special characters in filenames.
- **EcUtPerfData** -- Log Performance Data.

library EcUtNoPF

/ecs/formal/COMMON/CSCI_Util/src/Logging

EcUtNoPF is a utility library for all the ECS applications that do not use Process Framework. It provides following services:

- Error Logging Mechanism
- Passing and returning status code

- Resuable interface for objects which need to be transferred between processes
- Provides a unique identifier that is guaranteed to be unique within the ECS system.
- Uniquely identifies any RPC call between a client and a server process anywhere in the ECS enterprise.

Classes

- **EcUtErrorDetails** -- EcUtErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- **EcUtRpcID** -- EcUtRpcID is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- **EcUtStreamable** -- The EcUtStreamable class provides an reuseable interface for ÿ// objects which need to be transferred between processes.
- **EcUtUniqueID**

library EcUtSh (shared)

/ecs/formal/COMMON/CSCI_Util/src/Logging

EcUt is a utility library for all the ECS applications. It provides following services:

- Error Logging Mechanism
- Passing and returning status code
- Resuable interface for objects which need to be transferred between processes
- Provides a unique identifier that is guaranteed to be unique within the ECS system.
- Uniquely identifies any RPC call between a client and a server process anywhere in the ECS enterprise.

Classes

- **EcLgErrorEvent** -- This object provides a simple mechanism for converting a light-weight uuid_s_ok error message object (EcLgErrorMsg) into an EcAgEvent.
- **EcLgErrorMsg** -- This object encapsulates an ECS error event in a more lightweight contain than the EcAgEvent.
- **EcUtErrorDetails** -- EcUtErrorDetails is an accessor class for obtaining detailed information about an error code received from downstream.
- **EcUtRpcID** -- EcUtRpcID is used to uniquely identify any RPC call between a client and a server process anywhere in the ECS enterprise.
- **EcUtStreamable** -- The EcUtStreamable class provides an reuseable interface for ÿ// objects which need to be transferred between processes.
- **EcUtUniqueID**

library event

/ecs/formal/CSS/DOF/src/LOGGING/clientlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- **EcUtClientLogger** -- EcUtClientLogger is used for Application logs.
- **EcUtLoggerRelA**

library eventSh (shared)

/ecs/formal/CSS/DOF/src/LOGGING/clientlog

Purpose of this library is to provide a framework for MSS Agent to store information to the MSS log file and rename the file when the file is full.

Classes

- **EcUtClientLogger** -- EcUtClientLogger is used for Application logs.
- **EcUtLoggerRelA**

library Gl

/ecs/formal/DSS/sdsrv/src/gl

This library is used by many subsystems to provide a general purpose list object for storing various scalar and complex datatypes. The list and its elements can be read from or saved to a stream or file. This facilitates exchange of information (parameters) between processes.

Classes

- **GlBinaryP**
- **GlCircleP**
- **GlClient**
- **GlDateP** -- Specialization of GlParameter for dates.
- **GlDoubleP** -- Specialization of GlParameter for doubles.
- **GlGPolygonP** -- GlGPolygonP::GetBoundary.
- **GlLog** -- Place-holder for full-fledged log.
- **GlLongP** -- Specialization of GlParameter for type long.
- **GlParameter** -- Declaration of GlParameter (to provide type-safe casting, add a virtual X* GetX(); for every derived type. (eventually, of course, RTTI will make all GetX() functions obsolete...)..

- **GlParameterBase** -- This base class is the real RWCollectable, but since the RW macros don't allow pure virtual functions, GlParameter inherits from this and is not, in fact, a truly macro-ized RWCollectable. This is ok because no GlParameter object will ever be (only
- **GlParameterList**
- **GlPfxstream**
- **GlPfxstreambuf** -- Specialized streambuf that adds the start-of-line state and emits the given prefix at the beginning of each line.
- **GlPoint** -- Light weight data structure for spatial 2D point.
- **GlPointP**
- **GlPointSet**
- **GlPolygon** -- Light weight data structure for a polygon derived from GlPointSet
- **GlPolygonP**
- **GlPolygonSet**
- **GlPolygonSetP** -- PolygonSet as GlParameter.
- **GlRectangleP**
- **GlSpatialP**
- **GlStringP** -- Specialization of GlParameter for strings.
- **GlTimeP** -- Specialization of GlParameter for time data.
- **GlTransfer** -- This llass contains the specialized Flatten and Restore operations for RWCollectobles, which are necessary to portably send them across the network to other machines with potentially different architectures.

library GlSh (shared)

/ecs/formal/DSS/sdsrv/src/gl

This library is used by many subsystems to provide a general purpose list object for storing various scalar and complex datatypes. The list and its elements can be read from or saved to a stream or file. This facilitates exchange of information (parameters) between processes.

Classes

- **GlBinaryP**
- **GlCircleP**
- **GlClient**
- **GlDateP** -- Specialization of GlParameter for dates.
- **GlDoubleP** -- Specialization of GlParameter for doubles.
- **GlGPolygonP** -- GlGPolygonP::GetBoundary.
- **GlLog** -- Place-holder for full-fledged log.

- **GlLongP** -- Specialization of GlParameter for type long.
- **GlParameter** -- Declaration of GlParameter (to provide type-safe casting, add a virtual X* GetX(); for every derived type. (eventually, of course, RTTI will make all GetX() functions obsolete...)..
- **GlParameterBase** -- This base class is the real RWCollectable, but since the RW macros don't allow pure virtual functions, GlParameter inherits from this and is not, in fact, a truly macro-ized RWCollectable. This is ok because no GlParameter object will ever be (only
- **GlParameterList**
- **GlPfxstream**
- **GlPfxstreambuf** -- Specialized streambuf that adds the start-of-line state and emits the given prefix at the beginning of each line.
- **GlPoint** -- Light weight data structure for spatial 2D point.
- **GlPointP**
- **GlPointSet**
- **GlPolygon** -- Light weight data structure for a polygon derived from GlPointSet
- **GlPolygonP**
- **GlPolygonSet**
- **GlPolygonSetP** -- PolygonSet as GlParameter.
- **GlRectangleP**
- **GlSpatialP**
- **GlStringP** -- Specialization of GlParameter for strings.
- **GlTimeP** -- Specialization of GlParameter for time data.
- **GlTransfer** -- This llass contains the specialized Flatten and Restore operations for RWCollectobles, which are necessary to portably send them across the network to other machines with potentially different architectures.

library hdiff

/ecs/formal/PDPS/DPS/SSIT/src/DIF/HDIFF

library IcDarApi

/ecs/formal/CSS/DOF/src/RELB_GATEWAY/DAR/gds/api

This is the ASTER GDS client library which the DAR Communications Gateway links with to provide communication to the ASTER servers. This is not ECS developed source code, it was developed by GDS.

library InADPP

/ecs/formal/INGEST/ADPP/src/NMC

This library provides additional preprocessing functionality.

Classes

- **InGRIBData** -- This class defines the high level routines used in the decoding of GRIB data.
- **InPpGRIBHDF** -- Define the class for prepocessing GRIB HDF data.
- **InPpGRIBMetadata** -- Define the class for prepocessing GRIB metadata.
- **InPpGRIBScienceData** -- Define the class for preprocessing GRIB science data.

library InConfig

/ecs/formal/INGEST/common/src/config

This library provides a class which is used by each Ingest executable to obtain parameters from its configuration file.

Classes

- **EcUtConfig** -- The function opens the configuration file and allocates memory for the class variables.
- **InConfig** -- The InConfig class is used by each Ingest executable to obtain parameters from its configuration file.

library InDBaccUt

/ecs/formal/INGEST/common/src/DBaccess

This library provides classes which allows access to the Ingest database.

Classes

- **InCurrentDataTypeMap** -- The InCurrentDataTypeMap class provides a mapping of current versionID to valid Ingest data types.
- **InDataTypeTemplate** -- The InDataTypeTemplate class provides the valid Ingest data types.
- **InDBAccess** -- The InDBAccess class provides access to the Ingest database.
- **InDBAccessPool** -- The InDBAccessPool class is used by Ingest to obtain a connection to the Ingest database.
- **InEDPAddressMap** -- The InEDPAddressMap class provides a mapping of external data Provider with given IP address to valid Ingest externaldataprovider.
- **InESDTMap** -- The InESDTMap class provides a mapping of external data provider data type and data descriptor to valid Ingest data types.
- **InExternalDataProviderInfo** -- The InExternalDataProviderInfo class provides the valid Ingest external data provider information.
- **InFileTypeTemplate** -- The InFileTypeTemplate class provides the file type information for each data type.

- **InFileTypeTemplateError** -- The InFileTypeTemplateError class is the the error handler for the InFileTypeTemplate class.
- **InGranuleServerInfo** -- The InGranuleServerInfo class provides the Ingest Granule Server information.
- **InMediaType** -- The InMediaType class provides the Ingest media types.
- **InNextAvailableID** -- The InNextAvailableID class keeps track of and issues the next available number for RequestIDs.
- **InRequestFileInfo** -- The InRequestFileInfo class provides checkpoint storage of file information associated with a given data granule within a given Ingest request.
- **InRequestProcessData** -- The InRequestProcessData contains real-time process information regarding Data Types within a given request currently being processed or waiting to be processed by the system.
- **InRequestProcessHeader** -- The InRequestProcessHeader class contains real-time process information regarding Ingest requests currently being processed or waiting to be processed by the system.
- **InRequestSummaryData** -- The InRequestSummaryData class provides long-term storage of summary data type statistics associated with a given data granule in a given Ingest request.
- **InRequestSummaryHeader** -- The InRequestSummaryHeader class provides long-term storage of summary request-level statistics associated with a given Ingest request.
- **InSourceMCF** -- The InSourceMCF class contains metadata extraction information.
- **InSourceMCFError** -- The InSourceMCFError class is the error handler for the InSourceMCF class.
- **InSystemParameters** -- The InSystemParameters class provides the Ingest system-level tunable parameters.
- **InTDbDataTypeTemplate** -- Structure containing the data in row of the DataTypeTemplate table.
- **InTDbDDNFileInfo** -- Structure containing the DDN data needed for one row of the RequestFileInfo table.
- **InValDataGranuleState** -- The InValDataGranuleState class provides the valid data granule states.
- **InValNotifyType** -- The InValNotifyType class provides the valid notify types.
- **InValRequestState** -- The InValRequestState class provides the valid request states.

library InFTP

/ecs/formal/INGEST/common/src/ftp

This library provides a class with file transfer capabilities.

Classes

- **CsFtFTP** -- This Class provides file transfer capabilities.

library InGranuleC

/ecs/formal/INGEST/GranServer/src/Server

This library supports the client side of the SRF request.

Classes

- **InGranuleAsync_C** -- The InGranuleAsync_C object provides bi-directional communications with the InGranuleAsync_S object which resides in the server address space.
- **InGranuleList** -- This class contains granule information pertaining to a DAN request.

library InGranuleMsg

/ecs/formal/INGEST/GranServer/src/Server

This library supports the messages needed for SRF to communicate between Granule Server and Request Manager.

Classes

- **InGranuleCompleteMsg**
- **InGranuleRequestMsg**

library InGuiUt

/ecs/formal/INGEST/GUI/src/guiUtil

This library provides generic GUI capabilities.

Classes

- **InGUICommonRPUtil** -- This class provides common service used by the other GUI classes.
- **InHistoryLogRPUtil** -- This Class allows the operator to view (history) status of previous request.
- **InMediaIngestRPUtil** -- This all allows the operations to ingest media via the INGEST GUI.
- **InOperatorToolsRPUtil** -- This Class allows the operator to manipulate the INGEST (the external dataprovider table and system table) via the INGEST GUI.
- **InRequestControllerRPUtil**
- **InRequestMgrIF_1_0**

library InHostInfo

/ecs/formal/INGEST/common/src/hostinfo

Classes

- **EcUtHostInfo** -- This class encapsulate few funtion call in the form the socket library.

library InMetadataToolAPI

/ecs/formal/INGEST/ADPP/src/PGS

This library provides the **class** which accesses the PGS Toolkit from a non PGE C++ environment.

Classes

- **InMetadataToolAPI** -- This class encapsulates the TOOLKITS "c" language library call and extends the TOOLKIT functionality.

library InMsg

/ecs/formal/INGEST/common/src/Messages

This library provides classes that are used for all data messages exchanged between the external source and Ingest.

Classes

- **CsGwTransferPkt_1_0**
- **InDAN** -- The InDAN class contains services to access PVL statements in the DAN message.
- **InDDA** -- The InDDA class provides services to unpack the DDA message.
- **InDDN** -- The InDDN class provides services to package the DDN message.
- **InDDNTransferPkt_1_0**
- **InLongDAA** -- The InLongDAA class provides services to package the long DAA message.
- **InLongDDN** -- The InLongDDN class provides services to package the long DDN message.
- **InMessage** -- The InMessage class defines all data messages exchanged between the external source and Ingest.
- **InShortDAA** -- The InShortDAA class provides services to package the short DAA message.
- **InShortDDN** -- The InShortDDN class provides services to package the short DDN message.
- **PktStructDef_1_0**
- **SdpfTransferPkt_1_0**

library InPreprocess

/ecs/formal/INGEST/Preprocess/src/code2

This library provides preprocessing functions.

Classes

- **InBOMetadata**
- **InBUFRMetadata** -- This class provides the ability to preprocess parameter value metadata.
- **InDataPreprocessList** -- The class provides the ability to add files to a list and to get files off the list.
- **InDataPreprocessTask** -- The Class provides the ability to preprocess ingest files associated with various data types.
- **InDataPreprocessTaskError** -- Class that handles any error that occurred during preporocess.
- **InDataServerInsertionTask** -- The class provides the ability to insert data files into the Document Data Server and Science Data Server.
- **InEDOSMetadata** -- This class provides the ability to preprocess parameter value metadata.
- **InFDDMetadata** -- This class provides the ability to preprocess parameter value metadata.
- **InISCCPMetadata**
- **InMetadata** -- The InMetadata class is the abstract base class for other metadata classes such as the InBOMetadata class, InPVMetadata, and InHDFMetadata class.
- **InNCEPMetadata** -- This class provides the ability to preprocess parameter value metadata.
- **InNCEPT62Metadata** -- This class provides the ability to preprocess parameter value metadata.
- **InODLMetadata** -- The class file provides the ability to preprocess parameter ODL metadata.
- **InOZONSBMetadata**
- **InPpCCSDSTime** -- This class converts the CCSDS Spacecraft Time Stamp to the Ingest Preprocessing date time string format.
- **InPpConvertSOLib** -- Define the class to convert SO library.
- **InPpDate** -- Define the Ingest Preprocessing date class.
- **InPpDateTime** -- Define the class for Preprocessing date and time.
- **InPpDateTimeArray** -- Define the class for Preprocessing date and time arrays.
- **InPpExternalScienceData** -- Define the class for Ingest Preprocessing external science data.
- **InPpODLTools** -- Define the class for Preprocessing ODL Tools.
- **InPpPGSToolkitUtils** -- Define the class for the PGS toolkit utilities.
- **InPpScienceData** -- This class defines the base class InPpScienceData for various science data specializations.
- **InPpSharedObjectLibrary** -- Define the Ingest Prepocessing shared object library.
- **InPpTime** -- Define the Ingest Preprocessing time related stuff.
- **InPVMetadata** -- This class provides the ability to preprocess parameter value metadata.

- **InSpecMetadata** -- The class file provides the ability to preprocess parameter ODL metadata.
- **InVEGIMetadata** -- This class provides the ability to preprocess Third Generation Global Vegetation Indexmetadata.

library InResource

/ecs/formal/INGEST/Resource/src/resource

This library allocates resources for staging disk and IngestFtp.

Classes

- **IndataTypeList** -- The set of data types associate with the ingest files.
- **InFile** -- It transfers an ingested file.
- **InResourceIF** -- Serves as an interface to resource/device services (e.g., device allocation, data transferring, and file information) which is provided by the Data Server Subsystem.

library InRmQueue

/ecs/formal/INGEST/RequestMgr/src/Request

This library contains code for Request Managers queue.

Classes

- **InGranuleServersQueue** -- This class defines the singleton queue InGranuleServersQueue used by the Request Manager to keep track of pending requests for all available Granule Servers.
- **InPendingQueue** -- This class defines the queue InPendingQueue which is a priority queue of pending requests(implemented as InPendingQueueInfo) for a particular Granule Server.
- **InPendingQueueInfo** -- This class defines the objects which make up the InPendingQueue queue.

library InSnowIce

/ecs/formal/INGEST/ADPP/src/SnowIce

This library provides the class which inserts all of the data sets extracted from the native format file into SDSs within HDF-EOS.

Classes

- **InSnowIceData** -- The InSnowIceData class provides the operations for extracting and reformatting the data from a NESDIS EDR Masterfile.
- **InSnowIceHDFMgr** -- The InSnowIceHDFMgr class inserts all of the data sets extracted from the native format file into SDSs with HDF-EOS.

library InSysCmd

/ecs/formal/INGEST/common/src/syscmd

This library provies classes which allows applications to invoke multi-line unix commands.

Classes

- **EcUtSysCmd** -- Helps execute multiline system commands using sh.

library InTOMS

/ecs/formal/INGEST/ADPP/src/TOMS

This library provides the class that is designed to pre-process TOMS data in CD-ROM format into the standard ECS Internal format.

Classes

- **InTOMSData** -- This class is used to process TOMS data.
- **InTOMSHDF** -- This class is used to process TOMS HDF data.

library InUpdateUR

/ecs/formal/INGEST/common/src/ur

This library provides definitions and prototypes for **InUpdateUR** class.

Classes

- **InUpdateUR** -- This Class interfaces with Advertising Services to get the UR associated with each Data type.

library IoAdCore

/ecs/formal/IOS/AdvService/src/CoreLib

The CoreLib CSC provides a set of functionalities for describing a generic advertisement, parsing and validating attributes that support the ECS Core meta data model, and requesting and moderating advertisements

CoreLib provides a set of API for users/contributors to submit requests. There are several types of requests such as addition of new advertisements, modification or deletion of an already existing approved ads. It also provides a set of API for moderators to fetch, store, accept, reject, or save incoming requests. And finally, it provides a set of API for moderation group administration such as fetching, storing, updating, or deleting group. The use of moderator and administrator require proper DCE ACL privileges.

Classes

- **IoAdAdvertisement** -- This abstract base entity class supports operations to allow the defintion of a general advertisement entity.

- **IoAdAdvMessage** -- This class supports remote execution of operations of objects derived from IoAdAdvertisement objects.
- **IoAdApprovedAdv** -- This class models the baseline of approved ads.
- **IoAdApprovedAdvList** -- This class behaves as a list of approved ads with two additional features.
- **IoAdApprovedAdvProxy** -- This class aids in the implementation of a approved advertisement.
- **IoAdApprovedAdvRep** -- This class provides DCE-server based persistent operations for an approved advertisement.
- **IoAdAttribute** -- Store attributes associated with an advertisement.
- **IoAdAttributeList** -- We are persistent list of attributes with two additional features.
- **IoAdAttributeOperations** -- Defines member functions for IoAdAttributeOperations class that support submission and searching of new data types.
- **IoAdClassID** -- This class acts as a convertor between the standard classID and our needs.
- **IoAdConfigFile** -- This class is used to store all the sections of the config file into the dictionary file.
- **IoAdConfigSection** -- This class is using to store all the parameters of a section in the config file.
- **IoAdDescParser** -- Parser utility that reads HTML files in order to generate attribute lists required by Advertisements.
- **IoAdFactory** -- This class will generate the appropriate objects based on the type.
- **IoAdGroup** -- Advertisements that are logically related by subject area are all placed in a IoAdGroup, and a given moderator is assigned to moderate all ads in that group.
- **IoAdGroupProxy** -- This class aids in the implementation of a group of related advertisements.
- **IoAdGroupRep** -- This class provides DCE-server based persistent operations for a group of advertisements.
- **IoAdGroupState** -- Embody the state and state-related operations for a group of advertisements.
- **IoAdParseEvent** -- This class models an event that occurs while "parsing".
- **IoAdParseEventList** -- A list of events that occurred while parsing complex data.
- **IoAdRequest** -- This class models all requests made of advertising to change the baseline of approved ads.
- **IoAdRequestMessage** -- We extend the locale processing to handle Accept and Reject operations
- **IoAdRequestProxy** -- This class aids in the implementation of a request to change the baseline of approved advertisements.

- **IoAdRequestRep** -- This class provides DCE-server based persistent operations for a request to change the set of approved advertisements.
- **IoAdRequestState** -- Embody the state and state-related operations for a request to add, update or delete an approved ad (IoAdApprovedAdv).

library IoAdCoreSh (shared)

/ecs/formal/IOS/AdvService/src/CoreLib

The CoreLib CSC provides a set of functionalities for describing a generic advertisement, parsing and validating attributes that support the ECS Core meta data model, and requesting and moderating advertisements

CoreLib provides a set of API for users/contributors to submit requests. There are several types of requests such as addition of new advertisements, modification or deletion of an already existing approved ads. It also provides a set of API for moderators to fetch, store, accept, reject, or save incoming requests. And finally, it provides a set of API for moderation group administration such as fetching, storing, updating, or deleting group. The use of moderator and administrator require proper DCE ACL privileges.

Classes

- **IoAdAdvertisement** -- This abstract base entity class supports operations to allow the defintion of a general advertisement entity.
- **IoAdAdvMessage** -- This class supports remote execution of operations of objects derived from IoAdAdvertisement objects.
- **IoAdApprovedAdv** -- This class models the baseline of approved ads.
- **IoAdApprovedAdvList** -- This class behaves as a list of approved ads with two additional features.
- **IoAdApprovedAdvProxy** -- This class aids in the implementation of a approved advertisement.
- **IoAdApprovedAdvRep** -- This class provides DCE-server based persistent operations for an approved advertisement.
- **IoAdAttribute** -- Store attributes associated with an advertisement.
- **IoAdAttributeList** -- We are persistent list of attributes with two additional features.
- **IoAdAttributeOperations** -- Defines member functions for IoAdAttributeOperations class that support submission and searching of new data types.
- **IoAdClassID** -- This class acts as a convertor between the standard classID and our needs.
- **IoAdConfigFile** -- This class is used to store all the sections of the config file into the dictionary file.
- **IoAdConfigSection** -- This class is using to store all the parameters of a section in the config file.
- **IoAdDescParser** -- Parser utility that reads HTML files in order to generate attribute lists required by Advertisements.

- **IoAdFactory** -- This class will generate the appropriate objects based on the type.
- **IoAdGroup** -- Advertisements that are logically related by subject area are all placed in a IoAdGroup, and a given moderator is assigned to moderate all ads in that group.
- **IoAdGroupProxy** -- This class aids in the implementation of a group of related advertisements.
- **IoAdGroupRep** -- This class provides DCE-server based persistent operations for a group of advertisements.
- **IoAdGroupState** -- Embody the state and state-related operations for a group of advertisements.
- **IoAdParseEvent** -- This class models an event that occurs while "parsing".
- **IoAdParseEventList** -- A list of events that occurred while parsing complex data.
- **IoAdRequest** -- This class models all requests made of advertising to change the baseline of approved ads.
- **IoAdRequestMessage** -- We extend the locale processing to handle Accept and Reject operations
- **IoAdRequestProxy** -- This class aids in the implementation of a request to change the baseline of approved advertisements.
- **IoAdRequestRep** -- This class provides DCE-server based persistent operations for a request to change the set of approved advertisements.
- **IoAdRequestState** -- Embody the state and state-related operations for a request to add, update or delete an approved ad (IoAdApprovedAdv).

library IoAdHtmlCoreSh (shared)

/ecs/formal/IOS/AdvService/src/HtmlCoreLib

Classes

- **IoAdEsodamPage** -- This class creates an esod page.
- **IoAdEsodamPageRep** -- This class creates an esod page.
- **IoAdEsodPage** -- This class creates an esod page.
- **IoAdEsodPageRep** -- This class creates an esod page.
- **IoAdGroupSelection** -- This class encapsulates HTML user interface views to selecting an exisiting group.
- **IoAdPage** -- This class creates an advertisement page.
- **IoAdPageRep** -- This class creates an advertisement page; The constructor sets the contact and mail address.
- **IoAdSelection** -- This class encapsulates HTML user interface views to selecting an exisiting advertisement.

library IoAdHtmlSubsSh (shared)

/ecs/formal/IOS/AdvService/src/HtmlSubsLib

Classes

- **IoAdAdvertisementView** -- This abstract class encapsulates providing a user interface for the IoAdAdvertisement class.
- **IoAdApprovedAdvView** -- HTML user interface views for creating/updating/displaying Approved Adv.
- **IoAdGroupView** -- This class encapsulates providing a user interface for the IoAdGroup class.
- **IoAdInstallableServiceAdvView** -- This class encapsulates providing a user interface for the IoAdInstallable class.
- **IoAdMimeServiceView** -- This class encapsulates providing a user interface for the IoAdMimeServiceAdv class.
- **IoAdProductView** -- This class encapsulates providing a user interface for the IoAdProduct class.
- **IoAdProviderView** -- This class encapsulates providing a user interface for the IoAdProvider class.
- **IoAdRequestView** -- This class encapsulates providing a user interface for the IoAdRequest class.
- **IoAdServerInstall** -- This class supports handling all Web Server side tasks for installing MIME and Installable services.
- **IoAdServiceView** -- This class encapsulates providing a user interface for the IoAdService class.
- **IoAdSignatureServiceView** -- This class encapsulates providing a user interface for the IoAdSignature ServiceAdv class.
- **IoAdViewBuilder** -- This class supports operations to build advertisement views.

library IoAdInstallBase

/ecs/formal/IOS/AdvService/src/InstallLib

The installer allows one to install a compressed tar file containing some kind of application (modelled through a "IoAdInstallableServiceAdv") or a URL to get to a WWWW service (modelled through a "IoAdMimeServiceAdv") onto a users workstation.

When the above mentioned ads are displayed to the user, they contain a hyperlink to a CGI bin program, "IoAdInstallForm". This program then displays a form to allow the user to customize the install process. When the form is complete, the same IoAdInstallForm prgram is called to process the form. This program requests the Web browser to setup a workstation process "IoAdInstaller" and the server process sends relevant data to the workstation process. The workstation process then can FTP the tar file if this is an application.

In any case the workstation process then calls the ECS Installer developed in the CLS subsystem to actually do the install.

This library contains classes and programs to support the above activities.

Classes

- **IoAdFtpWrapper** -- This class may be used for automating the interactive program ftp -- the internet standard for file transfer.
- **IoAdInstallState** -- Class thats contains the shared state between the installer on the workstation and the installer on the server.

library IoAdInstallBaseSh (shared)

/ecs/formal/IOS/AdvService/src/InstallLib

The installer allows one to install a compressed tar file containing some kind of application (modelled through a "IoAdInstallableServiceAdv") or a URL to get to a WWWW service (modelled through a "IoAdMimeServiceAdv") onto a users workstation.

When the above mentioned ads are displayed to the user, they contain a hyperlink to a CGI bin program, "IoAdInstallForm". This program then displays a form to allow the user to customize the install process. When the form is complete, the same IoAdInstallForm prgram is called to process the form. This program requests the Web browser to setup a workstation process "IoAdInstaller" and the server process sends relevant data to the workstation process. The workstation process then can FTP the tar file if this is an application.

In any case the workstation process then calls the ECS Installer developed in the CLS subsystem to actually do the install.

This library contains classes and programs to support the above activities.

Classes

- **IoAdFtpWrapper** -- This class may be used for automating the interactive program ftp -- the internet standard for file transfer.
- **IoAdInstallState** -- Class thats contains the shared state between the installer on the workstation and the installer on the server.

library IoAdProxy

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This is the client side library.

Classes

- **IoAdConfiguration** -- This class provides access to configuration parameters for this subsystem.

- **IoAdIdentifiedProxy** -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- **IoAdIdentifiedServer** -- This class enables proxy/server persistence processing for classes with IDs.
- **IoAdLiteClient** -- This class supports operations that encapuslates becoming a Client-ProcessFramework-Process.
- **IoAdManagedServer** -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- **IoAdMessage** -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- **IoAdPersistentMessage** -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- **IoAdRpc_1_0**
- **IoAdRpc_1_0_ABS**
- **IoAdRpc_1_0_Mgr**
- **IoAdSession** -- This class provides access to specific global objects used by the advertising Core library.

library IoAdProxySh (shared)

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This is the client side library.

Classes

- **IoAdConfiguration** -- This class provides access to configuration parameters for this subsystem.
- **IoAdIdentifiedProxy** -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- **IoAdIdentifiedServer** -- This class enables proxy/server persistence processing for classes with IDs.
- **IoAdLiteClient** -- This class supports operations that encapuslates becoming a Client-ProcessFramework-Process.
- **IoAdManagedServer** -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- **IoAdMessage** -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- **IoAdPersistentMessage** -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.

- **IoAdRpc_1_0**
- **IoAdRpc_1_0_ABS**
- **IoAdRpc_1_0_Mgr**
- **IoAdSession** -- This class provides access to specific global objects used by the advertising Core library.

library IoAdSearch

/ecs/formal/IOS/AdvService/src/SearchLib

The SearchLib CSC provides a set of searching capabilities for approved advertisements, for the administration of groups, and for moderation of requests. It contains a set of API to perform title, text, attributes, date, provider, or group search on approved advertisements.

Classes

- **IoAdApprovedAdvSearchCommand** -- This class provides interfaces for application to search the set of approved advertisements by specifying criterion.
- **IoAdAttrSearchCommand** -- This class provides interfaces for application to search the attributes that we using to create the dynamic html page.
- **IoAdGroupList** -- This class behaves as a list of groups that can stream itself.
- **IoAdGroupSearchCommand** -- This class provides interfaces for application to search the groups.
- **IoAdRequestSearchCommand** -- This class provides interfaces for applications to search for requests to change the approved advertisement baseline.
- **IoAdSearch** -- This class acts as both the client and the server for search related operations.
- **IoAdSearchMessage** -- This class supports remote execution of operations of objects derived from IoAdSearch.

library IoAdSearchSh (shared)

/ecs/formal/IOS/AdvService/src/SearchLib

The SearchLib CSC provides a set of searching capabilities for approved advertisements, for the administration of groups, and for moderation of requests. It contains a set of API to perform title, text, attributes, date, provider, or group search on approved advertisements.

Classes

- **IoAdApprovedAdvSearchCommand** -- This class provides interfaces for application to search the set of approved advertisements by specifying criterion.
- **IoAdAttrSearchCommand** -- This class provides interfaces for application to search the attributes that we using to create the dynamic html page.
- **IoAdGroupList** -- This class behaves as a list of groups that can stream itself.

- **IoAdGroupSearchCommand** -- This class provides interfaces for application to search the groups.
- **IoAdRequestSearchCommand** -- This class provides interfaces for applications to search for requests to change the approved advertisement baseline.
- **IoAdSearch** -- This class acts as both the client and the server for search related operations.
- **IoAdSearchMessage** -- This class supports remote execution of operations of objects derived from IoAdSearch.

library IoAdServer

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This library is the server side library.

Classes

- **IoAdConfiguration** -- This class provides access to configuration parameters for this subsystem.
- **IoAdIdentifiedProxy** -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- **IoAdIdentifiedServer** -- This class enables proxy/server persistence processing for classes with IDs.
- **IoAdLiteClient** -- This class supports operations that encapuslates becoming a Client-ProcessFramework-Process.
- **IoAdManagedServer** -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- **IoAdMessage** -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- **IoAdPersistentMessage** -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- **IoAdRpc_1_0**
- **IoAdRpc_1_0_ABS**
- **IoAdRpc_1_0_Mgr**
- **IoAdSession** -- This class provides access to specific global objects used by the advertising Core library.

library IoAdServerSh (shared)

/ecs/formal/IOS/AdvService/src/ServerLib

The ServerLib CSC provides classes for process framework DCE server and client. In additions, it contains classes for making RPC call to the server from the client. This library is the server side library.

Classes

- **IoAdConfiguration** -- This class provides access to configuration parameters for this subsystem.
- **IoAdIdentifiedProxy** -- Client processes of the IoAd server contain objects that send messages over the network to perform thier persistence operations.
- **IoAdIdentifiedServer** -- This class enables proxy/server persistence processing for classes with IDs.
- **IoAdLiteClient** -- This class supports operations that encapuslates becoming a Client-ProcessFramework-Process.
- **IoAdManagedServer** -- Class extends the standard managed process-framework server to include requirements for the IoAdServer.
- **IoAdMessage** -- This is the abstract base class for self addressed synchronous client messages that execute themselves on the server.
- **IoAdPersistentMessage** -- This class supports remote execution of any of the persistent operations of objects derived from EcPoPersistent.
- **IoAdRpc_1_0**
- **IoAdRpc_1_0_ABS**
- **IoAdRpc_1_0_Mgr**
- **IoAdSession** -- This class provides access to specific global objects used by the advertising Core library.

library IoAdSubs

/ecs/formal/IOS/AdvService/src/SubsLib

The SubsLib CSC provides a set of derived classes which represent different types of advertisements: products, services, and providers. These classes provide operations to delete, update, insert, and search advertisements.

Classes

- **IoAdInstallableServiceAdv** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a downloadable service.
- **IoAdMimeServiceAdv** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a mime service.

- **IoAdProduct** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a data product.
- **IoAdProvider** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a data/service provider.
- **IoAdService** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a service.
- **IoAdSignatureServiceAdv** -- This entity class supports operations to allow the definition, storage and retrevial of a advertisement of a signature service.

library IoAdSubsSh (shared)

/ecs/formal/IOS/AdvService/src/SubsLib

The SubsLib CSC provides a set of derived classes which represent different types of advertisements: products, services, and providers. These classes provide operations to delete, update, insert, and search advertisements.

Classes

- **IoAdInstallableServiceAdv** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a downloadable service.
- **IoAdMimeServiceAdv** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a mime service.
- **IoAdProduct** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a data product.
- **IoAdProvider** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a data/service provider.
- **IoAdService** -- This entity class supports operations to allow the defintion, storage and retrevial of a advertisement of a service.
- **IoAdSignatureServiceAdv** -- This entity class supports operations to allow the definition, storage and retrevial of a advertisement of a signature service.

library MD5Encrypt

/ecs/formal/COMMON/CSCI_Util/src/md5

library MsAcClnt

/ecs/formal/MSS/MCI/Ac/src

The Accountability Management Service makes the user profile available to the various subsystems, such as the Data Server subsystem, information such as the user's electronic mail address and the shipping address, which are used for the distribution of data products ordered. This library provide the APIs to other subsystems.

Classes

- **EcAcOrderCMgr** -- This class represents a client side manager request to create, retrieve, update, delete an order.
- **EcAcOrderMgr_1_0**
- **EcAcProfileMgr** -- This class represents client side manager for user profile which submit a request to retieve, insert, update delete a user profile.
- **EcAcRegUserMgr** -- This class represents client manager for a registered user which submit request to retrieve, create, update, delete a register user.
- **EcAcUsrRequestMgr** -- This class represents the client side manager to submit a request to MsAcRegUserMgr to retrieve, insert, update, delete a user request.
- **MsAcManagedClient**
- **MsAcRegUserMgr_1_0**
- **MsAcUsrProfileMgr_1_0**
- **MsAcUsrRequestMgr_1_0**

library MsAcClntSh (shared)

/ecs/formal/MSS/MCI/Ac/src

The Accountability Management Service makes the user profile available to the various subsystems, such as the Data Server subsystem, information such as the user's electronic mail address and the shipping address, which are used for the distribution of data products ordered. This library provide the APIs to other subsystems.

Classes

- **EcAcOrderCMgr** -- This class represents a client side manager request to create, retrieve, update, delete an order.
- **EcAcOrderMgr_1_0**
- **EcAcProfileMgr** -- This class represents client side manager for user profile which submit a request to retieve, insert, update delete a user profile.
- **EcAcRegUserMgr** -- This class represents client manager for a registered user which submit request to retrieve, create, update, delete a register user.
- **EcAcUsrRequestMgr** -- This class represents the client side manager to submit a request to MsAcRegUserMgr to retrieve, insert, update, delete a user request.
- **MsAcManagedClient**
- **MsAcRegUserMgr_1_0**
- **MsAcUsrProfileMgr_1_0**
- **MsAcUsrRequestMgr_1_0**

library MsAcComm

/ecs/formal/MSS/MCI/Ac/src

This library contains class used for both Accountability Server and Client

Classes

- **EcAcOrder** -- This class represents an EcAcOrder class and get/set method for each attributes.
- **EcAcOrderList**
- **MsAcAddress** -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- **MsAcError** -- This class is used as an exception object.
- **MsAcPathFinder** -- This function builds the path and filename of accountability related files and CDS names.
- **MsAcRegUser**
- **MsAcRegUserList**
- **MsAcSybaseTime** -- This class get time and date string that can insert into sybase date format.
- **MsAcUsrName** -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- **MsAcUsrProfile** -- This class represents the user profile information.
- **MsAcUsrProfileList**
- **MsAcUsrRequest** -- This class represents the user request information.
- **MsAcUsrRequestList**
- **MsAcUsrResUsage** -- This class stores system resource usage information for a registered user.

library MsAcCommSh (shared)

/ecs/formal/MSS/MCI/Ac/src

This library contains class used for both Accountability Server and Client

Classes

- **EcAcOrder** -- This class represents an EcAcOrder class and get/set method for each attributes.
- **EcAcOrderList**
- **MsAcAddress** -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- **MsAcError** -- This class is used as an exception object.
- **MsAcPathFinder** -- This function builds the path and filename of accountability related files and CDS names.

- **MsAcRegUser**
- **MsAcRegUserList**
- **MsAcSybaseTime** -- This class get time and date string that can insert into sybase date format.
- **MsAcUsrName** -- This class represents an aggregate of the MsAcUsrProfile and MsAcOrder.
- **MsAcUsrProfile** -- This class represents the user profile information.
- **MsAcUsrProfileList**
- **MsAcUsrRequest** -- This class represents the user request information.
- **MsAcUsrRequestList**
- **MsAcUsrResUsage** -- This class stores system resource usage information for a registered user.

library MsAcSrv

/ecs/formal/MSS/MCI/Ac/src

This library contains the classes used by Accountability Servers

Classes

- **EcAcOrderDB** -- This class is an abstraction of the Order and Request Database.
- **EcAcOrderManagedServer**
- **EcAcOrderMgr_1_0_ABS**
- **EcAcOrderMgr_1_0_Mgr**
- **EcAcOrderP** -- This class contains information access to table EcAcOrder.
- **EcAcRequestP** -- This class contain's EcAcRequest information to access database table EcAcRequest..
- **MsAcAddressP** -- This class contains information to access database table.
- **MsAcDbErrorHandler**
- **MsAcDCEContext**
- **MsAcDCEErr**
- **MsAcDCEPassword**
- **MsAcDCEPasswordChange**
- **MsAcDCEStdPassword**
- **MsAcDCEUserProfile**
- **MsAcDCEUserRgst**
- **MsAcError --** This class is used as an exception object.
- **MsAcManagedServer**

- **MsAcParms** -- This class is used to store runtime configuration data for Sybase.
- **MsAcRegUserDB** -- This class is an abstraction of the Registered User Database.
- **MsAcRegUserMgr_1_0_ABS**
- **MsAcRegUserMgr_1_0_Mgr**
- **MsAcRegUserP** -- This class contains information to access database table.
- **MsAcSystem** -- This class is used to initialize accountability.
- **MsAcUserIdParsing** -- This class is to pasing the userId in "myid.
- **MsAcUsrNameP** -- RWDBread operator overloading.
- **MsAcUsrProfile** -- This class represents the user profile information.
- **MsAcUsrProfileList**
- **MsAcUsrProfileMgr_1_0_ABS**
- **MsAcUsrProfileMgr_1_0_Mgr**
- **MsAcUsrProfileP** -- Out put from RWReader.
- **MsAcUsrRequestMgr_1_0_ABS**
- **MsAcUsrRequestMgr_1_0_Mgr**
- **MsAcUsrRequestP** -- This class contains method to access database table.
- **MsAcUsrResUsageP** -- This class contains method to acccess database table.

library MssStub

/ecs/formal/DM/ServerStubs/src

Classes

- **EcAcOrderCMgr** -- This class represents a client side manager request to create, retrieve, update, delete an order.
- **EcAcProfileMgr** -- This class represents client side manager for user profile which submit a request to retieve, insert, update delete a user profile.

library Persistant

/ecs/formal/DM/GatewayCSCI/src/Persistant

Classes

- **DmGwAttributeMap** -- It's public view consists of two constructors, one taking a collection id for the DDICT and the other taking a ShortName. In each case the other necessary primary key information (InfoMgrName) is retrieved using the configuration information for the .
- **DmGwDirectoryQuery** -- This class builds a SQL query using the input search parameters and submits it to the Data Dictionary database.
- **DmGwDirectoryResult** -- A container class that holds the results of a directory search.

- **DmGwKeywordSearch** -- This class contains the mechanism for extracting keywordNames from the Data Dictionary database for a given set of collectionId and siteId.
- **DmGwQuery** -- This is an interface class to the Sybase DB library.

library plac

/ecs/formal/PDPS/PLS/PLANG/src/PWB/ac/src

Activities

This library contains the classes of schedulable activities which include basic activity along with its derived objects, such as Disk Activity, Dpr, Pr and Ground Event activities.

Classes

- **PlAcActivity** -- An instance of a PlAcActivity represents a request to change the configuration of a resource.
- **PlAcDiskAct** -- Instances of PlAcDiskAct are activities that are directed specifically at the disk resource.
- **PlAcDpr** -- Instances of PlAcDpr are activities which represent Data Processing Requests.
- **PlAcForceLink** -- Instances of PlAcForceLink are dummy classes, used to trick the linker into linking classes which are usually not known at compile time.
- **PlAcGroundEvent** -- Instances of PlAcGroundEvent are activities which model ground events on resources.
- **PlAcPr** -- Instances of PlAcPr are complex activities which represent Production Requests.

library plcl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/cl/src

Client resource model (rm) messages, and noid.

This library contains inherited classes responsible for setting up client conmmunication with the resource model through specialized messages.

Classes

- **PlClAppl** -- A PlClAppl is an ECS resource model client application class It is responsible for setting up communication to a resource model and initializing the entire process.
- **PlClForceLink** -- Instances of PlClForceLink are used to force linkage of ECS resource model message classes which may not be present at link time.
- **PlClMsgAlloc** -- PlClMsgAlloc is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgAllocAct** -- PlClMsgAllocAct is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.

- **PlClMsgAllocCheck** -- PlClMsgAllocCheck is a SMsgAllocCheck request to check a specific resource to see if a specific activity on a specific plan over a specific interval can be allocated on that resource.
- **PlClMsgAllocMove** -- PlClMsgAllocMove is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgAllocRqst** -- PlClMsgAllocRqst is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlClMsgCol** -- Instances of PlClMsgCol are a message that carries collections from one agent to another.
- **PlClMsgGetActPool** -- Instances of PlClMsgGetActPool are messages requesting activities from the resource model.
- **PlClMsgGetActPoolRsp** -- Comprises a response to a request for activities, and carries a copy of all or part of the resource models activity pool.
- **PlClMsgGetPlans** -- Instances of PlClMsgGetPlans are a request to get a subset of plans from the resource model.
- **PlClMsgGetPlansRsp** -- Instances of PlClMsgGetPlansRsp are a response to a request sent by an PlClMsgGetPlans.
- **PlClMsgGetRsNms** -- PlClMsgGetRsNms is a request to get the names of all resources known by the resource model.
- **PlClMsgGetRsNmsRsp** -- PlClMsgGetRsNmsRsp contains the names of all resources known by the resource model.
- **PlClMsgGroundEvent** -- Instances of PlClMsgGroundEvent are messages that have an associated ground event.
- **PlClMsgPlan** -- Instances of PlClMsgPlan are messages that have an associated plan.
- **PlClMsgPlanDel** -- Instances of PlClMsgPlanDel are messages that tell the resource model to delete the carried plan from its pool.
- **PlClMsgPlanNew** -- Instances of PlClMsgPlanNew are a message that tells the resource model about a new plan that has been created.
- **PlClMsgPlanSave** -- Instances of PlClMsgPlanSave are messages which tell the resource model to save the currently carried plan in its resource pool.
- **PlClMsgPlanSaveAs** -- Instances of PlClMsgPlanSaveAs are messages which tell the resource model save the carried plan under another given name.
- **PlClMsgPlanUpdate** -- Instances of PlClMsgPlanUpdate are a message that tells the resource model that a plan has been updated.
- **PlClMsgVCol** -- Instances of PlClMsgVCol are a message that carries void collections from one agent to another.
- **PlClSrmNoid** -- Instances of PlClSrmNoid are HNoids that manage the finding, requests to, and receiving information from the system srm within the era process.

library PlCore1

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains the first half of Planning's core functionality. The CoreLib provides functionality for all areas of Planning. It was divided in half due to its large size.

Classes

- **PlConfig** -- This file defines a helper class to manage getting and setting parameters from the process framework configuration file.
- **PlDataGranInsOrdCollectable** -- The sole pupose of this class is to provide a way for other classes in this library to use the functionality of the RWGranuleCollectable classes without being derived from RWCollectable themselves.
- **PlDataGranule** -- This class describes an individual instance of an ESDT or data file within the PDPS, it may describe either predicted Data Granules, or Granules that are known to exist within the Science Data Server so that they may be.
- **PlDataGranuleCollection**
- **PlDataScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled on the condition that the optional output of a previous Dpr exist.
- **PlDataSource** -- PlDataSource is an abstract base class which declares attributes and function prototypes for all operations of PlDataSource and its subclasses.
- **PlDataSourceFactory** -- This class provides simple methods to construct the correct type of Data Source object from the persistent data in the database.
- **PlDataType** -- This class describes a Data Type known to the PDPS and is used to describe the input or output data of a PGE.
- **PlDataTypeReq** -- This class captures the attributes that describe the input relationship between a PGE and a Data Type A.
- **PlDbIfManager** -- This file defines a helper class to manage the various database interfaces.
- **PlDpr** -- This class describes a Data Processing Request.
- **PlDprData** -- This class associates Data-Granules with DPRs.
- **PlOutputYield** -- This class captures the attributes that describe the output relationship between a PGE and a Data Type The class is used when predicting the output data for the PGE that will be produced for a given Data Processing Request.
- **PlPerformance** -- This class describes the performance statistics of PGE.
- **PlPge** -- This class is the base class within a generalization heirachy that describes Product Generation Executives (PGEs) to the PDPS software, it defines the abstract operations required in order that Planning Subsystem can work out when a PGE requires to be
- **PlPgeFactory** -- This class provides simple methods to construct the correct type of PGE object from the persistent data in the database.
- **PlProductionRequest** -- A Production Request describes the need for the scheduling of some PGE to produce a series of Data Granules.

- **PlSnapshotScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.
- **PlTileScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period for a series of tiles.
- **PlTileSchema** -- The PlTileSchema class is a table used by PlTileScheduled for getting the spatial definitions of tiles.
- **PlTimer** -- This class initiates a timer to alert the subscription manager of the expiration of a wait time for alternat input activation.
- **PlTimeScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.

library PlCore1IF

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains the Science Data Server, advertisement, and subscription server interfaces that are needed to run the Production request editor.

Classes

- **PlConfig** -- This file defines a helper class to manage getting and setting parameters from the process framework configuration file.
- **PlDataGranInsOrdCollectable** -- The sole pupose of this class is to provide a way for other classes in this library to use the functionality of the RWGranuleCollectable classes without being derived from RWCollectable themselves.
- **PlDataGranule** -- This class describes an individual instance of an ESDT or data file within the PDPS, it may describe either predicted Data Granules, or Granules that are known to exist within the Science Data Server so that they may be.
- **PlDataGranuleCollection**
- **PlDataScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled on the condition that the optional output of a previous Dpr exist.
- **PlDataSource** -- PlDataSource is an abstract base class which declares attributes and function prototypes for all operations of PlDataSource and its subclasses.
- **PlDataSourceFactory** -- This class provides simple methods to construct the correct type of Data Source object from the persistent data in the database.
- **PlDataType** -- This class describes a Data Type known to the PDPS and is used to describe the input or output data of a PGE.
- **PlDataTypeReq** -- This class captures the attributes that describe the input relationship between a PGE and a Data Type A.
- **PlDbIfManager** -- This file defines a helper class to manage the various database interfaces.
- **PlDpr** -- This class describes a Data Processing Request.

- **PlDprData** -- This class associates Data-Granules with DPRs.
- **PlOutputYield** -- This class captures the attributes that describe the output relationship between a PGE and a Data Type The class is used when predicting the output data for the PGE that will be produced for a given Data Processing Request.
- **PlPerformance** -- This class describes the performance statistics of PGE.
- **PlPge** -- This class is the base class within a generalization heirachy that describes Product Generation Executives (PGEs) to the PDPS software, it defines the abstract operations required in order that Planning Subsystem can work out when a PGE requires to be
- **PlPgeFactory** -- This class provides simple methods to construct the correct type of PGE object from the persistent data in the database.
- **PlProductionRequest** -- A Production Request describes the need for the scheduling of some PGE to produce a series of Data Granules.
- **PlSnapshotScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.
- **PlTileScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period for a series of tiles.
- **PlTileSchema** -- The PlTileSchema class is a table used by PlTileScheduled for getting the spatial definitions of tiles.
- **PlTimer** -- This class initiates a timer to alert the subscription manager of the expiration of a wait time for alternat input activation.
- **PlTimeScheduled** -- This class is a specialization of the PlPge class and accounts for PGEs that are scheduled to process data for regular time period, this class accounts for all the PGEs that can be planned and scheduled in Release A.

library PlCore2

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains the second half of Planning's core functionality. The CoreLib provides functionality for all areas of Planning. It was divided in half due to its large size.

Classes

- **PlAlternate** -- This class defines an alternate data input to a PGE.
- **PlComputer** -- The PlComputer class, which is inherited from PlResource, .
- **PlDASDelta**
- **PlDASDifferent**
- **PlDATRecord** -- The PlDATRecord class maintains all the information the predicted availability times of data granules and some of the attributes of the data granules used to generate such predictions.
- **PlDbIfManager** -- This file defines a helper class to manage the various database interfaces.

- **PlDPRCollectionNB**
- **PlMetaDataChecks** -- These friend functions are needed by DB IF template.
- **PlNonRoutineArrival** -- This class provides a framework for handling data which arrives in a non-routine or random fashion.
- **PlOnDemandExceed**
- **PlOnDemandReplanValues**
- **PlPGECollection** -- This class handles collections of PGEs.
- **PlPgeDbQuery** -- This class provides functionality to handle specific queries to the PDPS database on behalf of PGEs.
- **PlPgePriority** -- This class is part of the production strategies which relates a particular PGE to a particular priority.
- **PlPrAlternateInputValues** -- This class defines an alternate input to a PGE.
- **PlPrMetaDataValues** -- These friend functions are needed by DB IF template.
- **PlProdStrat** -- This class describes the production strategies used by the DAAC to determine data processing request priorities.
- **PlProfile** -- This is a simple class to create a single instance of the client information class required by DSS when using any of it's client interfaces..
- **PlPRPriority** -- This class describes a table in the PDPS database that matches the types of production requests (on-demand, standard and reprocessing) and the priority assigned to requests of that type.
- **PlReplanCriteria**
- **PlRescUseThresh**
- **PlResource** -- This class is the base class in a generalization heirachy describing the production resource.
- **PlResourceChange** -- This class is used to compare an old resource plan with the current resource plan.
- **PlResourceRequirement**
- **PlRoutineArrival** -- Persistent attributes as defined in the schema.
- **PlUserParameters** -- This class allows for manipulation of the run time parameters that may be defined for a PGE.
- **PlUserParametersCollection** -- This class defines methods and operations to allow access to all of the User Parameters associated to a particular PGE.
- **PlUserPriority** -- This class is part of the production strategies which relates a particular user to a particular priority.

library pldb

/ecs/formal/PDPS/PLS/PLANG/src/PWB/db/src

Database Interface.

This library contains classes which build the database interface for ECS specific processes. Its classes are responsible for reading, writing from/to the database and construnting the corresponding objects.

Classes

- **PlDbDatabaseCfg** -- PlDbDatabaseCfg serves as a base class for all HCL database interface.
- **PlDbDatabaseInfo** -- PlDatabaseInfo serves as an encapuslation of the database schema and stores database login information.
- **PlDbDprCfg** -- PlDbDprCfg allows loading and saving data processing request (DPR) activities (PlAcDpr) from and to the database tables.
- **PlDbGroundEventCfg** -- PlDbGroundEventCfg allows loading and saving ground event activities (PlAcGroundEvent) from and to the database tables.
- **PlDbPlanCfg** -- PlDbPlanCfg allows loading and saving of Plan information (PlPlPlan) from and to the database tables.
- **PlDbPrCfg** -- PlDbPrCfg allows loading and saving Production Request activities (PlAcPr) from and to the database tables.
- **PlDbResourceCfg** -- PlDbResourceCfg allows loading the resource pool from the database tables and saving to the database tables from the resource pool.

library pldi

/ecs/formal/PDPS/PLS/PLANG/src/PWB/di/src

Common **Displayable** (motif) components.

This library contains inherited classes responsible for implementing display capabilities for the client application.

Classes

- **PlDiAppl** -- An instance of this class is a type of resource model client display application.
- **PlDiDateTimePanel** -- Instances of PlDiDateTimePanel are displayable panels of MOTIF widgets for the entry or display of date values in a "yy/ddd" format and time values in a "<hours>::<minutes>::<seconds> format. Seconds can be entered or displayed to 1 millisecond ..
- **PlDiMotifNotifier** -- Instances of PlDiMotifNotifier are Motif Notifiers for client display application which handle the watch set behavior.

library PlGUIError

/ecs/formal/PDPS/PLS/PLANG/src/GUIError

This library contains the objects needed for SSIT and QA Monitor GUIs to display error messages. PLS is not using this library.

Classes

- **DpAtMessageDialog**
- **EcsError**

library plip

/ecs/formal/PDPS/PLS/PLANG/src/PWB/ip/src

Interprocess Communication Agent.

This library contains classes which implement PLS communication among different processes.

Classes

- **PlIpAgent** -- Instances of PlIpAgent are HNoidAgents that have choosen a particular NameServAbs implementation, register their HAddress with said NameServAbs, and interact with the global HSigMgr to install a HNsSigHand to take over in the event of disaster.

library plmi

/ecs/formal/PDPS/PLS/PLANG/src/PWB/mi/src

Miscellaneous

This library contains miscellaneous classes that are useful to multiple processes. The class **PlMiAppl** in this library is the base class for all applications.

Classes

- **PlMiAppl** -- Instances of PlMiAppl are base application classes for all ECS applications.
- **PlMiTextConvertor** -- Instances of PlMiTextConvertor are functionoids responsible for string-to-binary and binary-to-string conversions of integer, floating and other binary values.
- **PlMiTimeCvtr** -- Instances of PlMiTimeCvtr provides additional time conversion operations.

library plpl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/pl/src

Plans.

This library represents the concepts of plans. Plans are collections of scheduled production activities on available resources.

Classes

- **PlPlForceLink** -- Instances of PlPlForceLink are used to force linkage of ECS plan classes which may not be present at link time.
- **PlPlPlan**

library plrc

/ecs/formal/PDPS/PLS/PLANG/src/PWB/rc/src

Resources.

This library contains ECS specific resource related classes. Resources represent the actual physical or virtual entities that perform the production runs. These resources are computers, disks, strings, etc. They maintain a representation of their state through time running state lists.

Classes

- **PlRcActIdFact** -- Instances of PlRcActIdFact are activity ID generators.
- **PlRcActState** -- Instances of PlRcActState model the state of a resource during which that resource is in the process of executing an activity.
- **PlRcAutosys** -- Resources to hold a place in the pool for the autosy(s).
- **PlRcComputer** -- Instances of PlRcComputer are multi-processors with a known configuration of associated disks.
- **PlRcCpu** -- Instances of PlRcCpu are processors associated with a particular computer and are of a certain type (Sun, SGI, etc.).
- **PlRcDataArrivalState** -- Instances of PlRcDataArrivalState models the current data arrival state of a resource.
- **PlRcDataServer** -- Instances of PlRcDataServer provide data server information.
- **PlRcDisk** -- Instances of PlRcDisk are computer resources capable of providing disk storage and disk capacity information.
- **PlRcForceLink** -- Instances of PlRcForceLink are used to force linkage of ECS resource model classes which may not be present at link time.
- **PlRcIdFactAbs** -- Instances of PlRcIdFactAbs are abstract base class id generators.
- **PlRcLocalDisk** -- Instances of PlRcLocalDisk are Disk storage units which are "Locally" or directly attached to a Computer and are therefore owned by that computer.
- **PlRcMultiActState** -- This class is a RRsState that can keep track of associations between it and multiple RActivity 'tasking' requests.
- **PlRcProcessState** -- Instances of PlRcProcessState are classes that represent the current processing state of the CPU.
- **PlRcResource** -- Instances of PlRcResource are classes from which all other Planning Workbench specific resource classes inherit.
- **PlRcResourceIdFact** -- Instances of PlRcResourceIdFact are activity ID generators.

- **PlRcState** -- Instances of PlRcState are generic activity states which simply contains the name of the generic state.
- **PlRcStorageState** -- Instances of PlRcStorageState are states for storage utilization for a disk.
- **PlRcString** -- Instances of PlRcString are logical groupings of computers.

library plrm

/ecs/formal/PDPS/PLS/PLANG/src/PWB/rm/src

Resource Model.

The resouce model is the heart of the Production Planning tool. It is a representation of database in memory, and the interface for reading and updating to the database, and a maintainer for an up-to-date representation of the states of resource, plans and activities in the system. It is also the hub of message switching process.

Classes

- **PlRmAppl** -- Instances of PlRmAppl are resource model applications.
- **PlRmCatalog** -- An instance of PlRmCatalog is owned by a PlRmSrm client.
- **PlRmSigHand** -- Instances of PlRmSigHand are signal catchers for a resource model to detect termination signals and save our resource model states.
- **PlRmSrm** -- Instances of PlRmSrm are resource model editors that know about ecs notions, plan accesses, etc.
- **PlRmSrmNoid** -- The PlRmSrmNoid is an ipc msgAgent that services resource model requests.

library PlRpAc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/ac

Manages the scheduling activities.

Classes

- **PlRpAcActivity** -- An instance of a PlRpAcActivity represents a request to change the configuration of a resource.
- **PlRpAcAlloc** -- An instance of a PlRpAcAlloc is an allocation containing information as to when the allocation was created, and any parameters that have been modified from the default parameters.
- **PlRpAcDpr** -- Instances of PlRpAcDpr are activities which represent Data Processing Requests.
- **PlRpAcForceLink** -- Instances of PlRpAcForceLink are dummy classes, used to trick the linker into linking classes which are usually not known at compile time.
- **PlRpAcGroundEvent** -- Instances of PlRpAcGroundEvent are activities which model ground events on resources.

- **PlRpAcInterval** -- This class defines attibutes and operations [get/set] for PlRpAcInterval.
- **PlRpAcResourceReservation** -- This class defines attibutes and operations for PlRpAcResourceReservation.
- **PlRpAcSchedInfo** -- Instances of PlRpAcSchedInfo are objects that contain information related to how an activity will/was scheduled.
- **PlRpAcTimeInfo** -- Instances of PlRpAcTimeInfo are base class objects for scheduling time information.

library PlRpCl

/ecs/formal/PDPS/PLS/PLANG/src/RPL/cl

Manages the resource model clients.

Classes

- **PlRpClAppl** -- A PlRpClAppl is an ECS resource model client application class It is responsible for setting up communication to a resource model and initializing the entire process.
- **PlRpClForceLink** -- Instances of PlRpClForceLink are used to force linkage of ECS resource model message classes which may not be present at link time.
- **PlRpClMsgAlloc** -- PlRpClMsgAlloc is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlRpClMsgAllocAct** -- PlRpClMsgAllocAct is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlRpClMsgAllocCheck** -- PlRpClMsgAllocCheck is a SMsgAllocCheck request to check a specific resource to see if a specific activity on a specific plan over a specific interval can be allocated on that resource.
- **PlRpClMsgAllocMove** -- PlRpClMsgAllocMove is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlRpClMsgAllocRqst** -- PlRpClMsgAllocRqst is a request to allocate a specific resource to a specific activity on a specific plan over a specific interval.
- **PlRpClMsgCol** -- Instances of PlRpClMsgCol are a message that carries collections from one agent to another.
- **PlRpClMsgGetActPool** -- Instances of PlRpClMsgGetActPool are messages requesting activities from the resource model.
- **PlRpClMsgGetActPoolRsp** -- Comprises a response to a request for activities, and carries a copy of all or part of the resource models activity pool.
- **PlRpClMsgGetPlans** -- Instances of PlRpClMsgGetPlans are a request to get a subset of plans from the resource model.
- **PlRpClMsgGetPlansRsp** -- Instances of PlRpClMsgGetPlansRsp are a response to a request sent by an PlRpClMsgGetPlans.

- **PlRpClMsgGetRsNms** -- PlRpClMsgGetRsNms is a request to get the names of all resources known by the resource model.
- **PlRpClMsgGetRsNmsRsp** -- PlRpClMsgGetRsNmsRsp contains the names of all resources known by the resource model.
- **PlRpClMsgPlan** -- Instances of PlRpClMsgPlan are messages that have an associated plan.
- **PlRpClMsgPlanNew** -- Instances of PlRpClMsgPlanNew are a message that tells the resource model about a new plan that has been created.
- **PlRpClMsgPlanSave** -- PlRpClMsgPlanSave is a request to save a plan to a file whose name is specified at the command line.
- **PlRpClMsgRsChg** -- Instances of PlRpClMsgRsChg are a message that tells the resource model about a modified resource.
- **PlRpClMsgRsDel** -- Instances of PlRpClMsgRsChg are a message that tells the resource model about a deleted resource.
- **PlRpClMsgRsNew** -- Instances of PlRpClMsgRsNew are a message that tells the resource model about a new resource.
- **PlRpClMsgVCol** -- Instances of PlRpClMsgVCol are a message that carries void collections from one agent to another.
- **PlRpClSrmNoid** -- Instances of PlRpClSrmNoid are HNoids that manage the finding, requests to, and receiving information from the system srm within the era process.

library PlRpDb

/ecs/formal/PDPS/PLS/PLANG/src/RPL/db

Accesses and modifies the database.

Classes

- **PlRpDbCommentsCfg** -- PlRpDbCommentsCfg class allows loading the comments for a resource from the database tables and saving the comment to the database tables from the GUI.
- **PlRpDbDatabaseCfg** -- PlRpDbDatabaseCfg class serves as a base class for all HCL database interface classes.
- **PlRpDbDatabaseInfo** -- PlDatabaseInfo class serves as an encapuslation of the database schema and stores database login information.
- **PlRpDbDprCfg** -- PlRpDbDprCfg allows loading and saving data processing request (DPR) activities (PlRpAcDpr) from and to the database tables.
- **PlRpDbGroundEventCfg** -- PlRpDbGroundEventCfg allows loading and saving ground event activities (PlRpAcGroundEvent) from and to the database tables.
- **PlRpDbMMSRsCfg** -- The PlRpDbMMSRsCfg class contains the control identification values from the configuration file received from Baseline Manager (MSS).

- **PlRpDbPlanCfg** -- PlRpDbPlanCfg allows loading and saving of Plan information (PlPlPlan) from and to the database tables.
- **PlRpDbReservationCfg** -- PlRpDbReservationCfg allows loading and saving reservation activities (PlRpAcResourceReservation) from and to the database tables.
- **PlRpDbResourceCfg** -- PlRpDbResourceCfg class allows loading the resource pool from the database tables and saving to the database tables from the resource pool.
- **PlRpDbRsCfg** -- The PlRpDbRsCfg class is the interface between Resource Planning and Baseline Manager (MSS).

library PlRpDi

/ecs/formal/PDPS/PLS/PLANG/src/RPL/di

A resource model client display application.

Classes

- **PlRpDiAppl** -- An instance of this class is a type of resource model client display application.

library PlRpRc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/rc

Manages resources.

Classes

- **PlRpRcActIdFact** -- Instances of PlRpRcActIdFact are activity ID generators.
- **PlRpRcActState** -- Instances of PlRpRcActState model the state of a resource during which that resource is in the process of executing an activity.
- **PlRpRcAllocIdFact** -- Instances of PlRpRcAllocIdFact are allocation ID generators.
- **PlRpRcAutosys** -- Instances of PlRpRcAutosys are logical groupings of strings.
- **PlRpRcComputer** -- Instances of PlRpRcComputer are multi-processors with a known configuration of associated disks.
- **PlRpRcCpu** -- Instances of PlRpRcCpu are processors associated with a particular computer and are of a certain type (Sun, SGI, etc.).
- **PlRpRcDisk** -- Instances of PlRpRcDisk are computer resources capable of providing disk storage and disk capacity information.
- **PlRpRcForceLink** -- Instances of PlRpRcForceLink are used to force linkage of ECS resource model classes which may not be present at link time.
- **PlRpRcIdFactAbs** -- Instances of PlRpRcIdFactAbs are abstract base class id generators.
- **PlRpRcLocalDisk** -- Instances of PlRpRcLocalDisk are Disk storage units which are "Locally" or directly attached to a Computer and are therefore owned by that computer.

- **PlRpRcMultiActState** -- This class is a RRsState that can keep track of associations between it and multiple RActivity 'tasking' requests.
- **PlRpRcNetworkDisk** -- Instances of PlRpRcNetworkDisk are Disk storage units which are "Network Attached" and behave like NFS mounted disks.
- **PlRpRcProcessState** -- Instances of PlRpRcProcessState are classes that represent the current processing state of the CPU.
- **PlRpRcRealComputer** -- Instance of PlRpRcRealComputer is a physical collection of virtual computers.
- **PlRpRcResource** -- Instances of PlRpRcResource are base class for string, computer and disk resources.
- **PlRpRcResourceIdFact** -- Instances of PlRpRcResourceIdFact are unique activity ID generators.
- **PlRpRcState** -- Instances of PlRpRcState are generic activity states which simply contains the name of the generic state.
- **PlRpRcStorageState** -- Instances of PlRpRcStorageState are states for storage utilization for a disk.
- **PlRpRcString** -- Instances of PlRpRcString are logical groupings of computers.

library PlRpRe

/ecs/formal/PDPS/PLS/PLANG/src/RPL/sre

Manages the Resource Editor GUI.

Classes

- **PlRpReAppl** -- Instances of PlRpReAppl are derived PlRpDiAppl's capable of using it's noid to request resources from the resource model.
- **PlRpReAutosysWin** -- An instance of this class provides a view of the details associated with the selected Autosys resource.
- **PlRpReComputerWin** -- An instance of this class provides a view of the details associated with the selected computer (host) resource.
- **PlRpReConfirmWin**
- **PlRpReDateDlg** -- An instance of the class brings up a date entry dialog for the MSS resource baseline.
- **PlRpReDiskWin** -- An instance of this class provides a view of the details associated with the selected disk partition resource.
- **PlRpReHardwareWin** -- An instance of this class provides a view of the details associated with the selected hardware resource.
- **PlRpReMsgBox** -- An instance of this provides a message popup box for displaying information to the user.

- **PlRpReNoid** -- Instances of PlRpReNoids handle resource planning related client noid messages.
- **PlRpReRealComputerWin** -- An instance of this class provides a view of the details associated with the selected realComputer resource.
- **PlRpReSelBox** -- An instance of this provides a message popup box for displaying information to the user.
- **PlRpReStringWin** -- An instance of this class provides a view of the details associated with the selected string resource.
- **PlRpReWin** -- This class is Motif Window for controlling the setup and display of the resource planning workbench.
- **PlRpReWinAbs** -- PlRpReWinAbs establishes protocol for System Resource Editor Windows.

library PlRpRm

/ecs/formal/PDPS/PLS/PLANG/src/RPL/srm

Manages the resource model.

Classes

- **PlRpRmAppl** -- Instances of PlRpRmAppl are resource model applications.
- **PlRpRmCatalog** -- An instance of PlRpRmCatalog is owned by a PlRpRmSrm client.
- **PlRpRmSigHand** -- Instances of PlRpRmSigHand are signal catchers for a resource model to detect termination signals and save our resource model states.
- **PlRpRmSrm** -- Instances of PlRpRmSrm are resource model editors that know about ecs notions, plan accesses, etc.
- **PlRpRmSrmNoid** -- The PlRpRmSrmNoid is an ipc msgAgent that services resource model requests.

library PlRpSc

/ecs/formal/PDPS/PLS/PLANG/src/RPL/sc

Manages schedulable resources.

Classes

- **PlRpScAllImpct** -- Instances of PlRpScAllImpct add the knowledge of oversubscription to the impact scheduler.
- **PlRpScComputer** -- Instances of PlRpScComputer are schedulable computer resources.
- **PlRpScCpu** -- Instances of PlRpScCpu are schedulable CPU resources.
- **PlRpScForceLink** -- Instances of PlRpScForceLink are used to force linkage of schedule resource model classes which may not be present at link time.
- **PlRpScLocalDisk** -- Instances of PlRpScLocalDisk are schedulable local disk resources.

- **PlRpScResource** -- This class sets up protocol for all ECS schedulable resources.

library PlRpSi

/ecs/formal/PDPS/PLS/PLANG/src/RPL/si

Manages the Resource Scheduler GUI.

Classes

- **PlRpSiActType** -- PlRpSiActType is the same of one row in ACTIVITY_TYPE table.
- **PlRpSiAppl** -- Instance of PlRpSiAppl is derived from PlRpDiAppl.
- **PlRpSiCloseWin**
- **PlRpSiEditWin** -- An instance of this class provides a view of the details associated with the selected reservation resource.
- **PlRpSiIntervalSelWin**
- **PlRpSiMsgBox** -- An instance of this provides a message popup box for displaying information to the user.
- **PlRpSiNoid** -- I'm a noid who is only interested in handling resource related SrmClient messages..
- **PlRpSiOppGen** -- Instances of PlRpSiOppGen are derived from SsiOppGen.
- **PlRpSiResourceSelWin**
- **PlRpSiScheduler** -- The PlRpSiScheduler is a generic Scheduling System user interface.
- **PlRpSiWin**
- **PlRpSiWinAbs** -- An instance of this class shares static attributes for all windows.

library PlRpTl

/ecs/formal/PDPS/PLS/PLANG/src/RPL/tl

Manages the scheduled Resources' Timeline GUI.

Classes

- **PlRpTlApDefsCfg** -- Instances of PlRpTlApDefsCfg handle configuration files for default display resource settings for the timeline.
- **PlRpTlAppl** -- PlRpTlAppl is the main application class for the timeline.
- **PlRpTlColChgList** -- Instances of PlRpTlColChgList are lists which are part of the color mapper dialog.
- **PlRpTlColChgSt** -- Instances of PlRpTlColChgSt map colors to types.
- **PlRpTlColChngr** -- Instances of PlRpTlColChngr can be used to specify colors for types.
- **PlRpTlConfig** -- PlRpTlConfig is the base class for all configuration files.

- **PlRpTlDefsConfig** -- Instances of PlRpTlDefsConfig load and save basic timeline configuration files.
- **PlRpTlFileDiag** -- An instance of the class is a base dialog for loading and saving the configuration files.
- **PlRpTlFileSelDiag** -- This is a derived DFileSelectBox that AUTOMATICALLY creates a FileSelectionDialog (i.e. both the FileSelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make
- **PlRpTlMainRegion** -- A PlRpTlMainRegion is a _TLMainRegion_ that implements "timeline" mouse tracking behavior.
- **PlRpTlPanel** -- This class, which is derived from TLPanel, adds capabilities for interactively changing colors and resources, and for saving and loading from different configuration files.
- **PlRpTlPlanWinChgr** -- Instances of PlRpTlPlanWinChgr are dialogs for selecting a plan.
- **PlRpTlRsConfig** -- Instances of PlRpTlRsConfig are responsible for loading a set of names of resources to be displayed on the timeline.
- **PlRpTlRsEditor** -- Instances of PlRpTlRsEditor are dialogs for choosing the set of resources to be displayed on the timeline.
- **PlRpTlSimpleEvent** -- Instances of PlRpTlSimpleEvent provide base class behavior that all timeline simple events will provide.
- **PlRpTlSliderPup** -- Instances of PlRpTlSliderPup are popup windows which display a start time and stop time for the corresponding plan region.
- **PlRpTlSrmNoid** -- Instances of PlRpTlSrmNoid are timeline resource model client noids.
- **PlRpTlStatusBar** -- Instances of PlRpTlStatusBar provide a line at the bottom of the timeline for displaying messages.
- **PlRpTlStplConfig** -- Instances of PlRpTlStplConfig handle stipple keywords in configuration files.
- **PlRpTlStplEv** -- Instances of PlRpTlStplEv are stipple events.
- **PlRpTlStplEvent** -- Instances of PlRpTlStplEvent are stipple events associated with resource states.
- **PlRpTlStplView** -- Instances of PlRpTlStplView are views associated with stipple events.
- **PlRpTlStplVw** -- Instances of PlRpTlStplVw are views of stipple events.
- **PlRpTlTypesConfig** -- Instances of PlRpTlTypesConfig handle configureation files which can contain class names which are added to the list of the timeline's known types..
- **PlRpTlValEvent** -- This class is a derived TLEvent which is used for displaying values versus time data graphically on a TLEventSubRegion over a specific interval of time.
- **PlRpTlValView** -- This class is the base class for views that display value versus time information graphically in a timeline subregion.
- **PlRpTlWindow** -- PlRpTlWindow are main windows used by the PlRpTlAppl class.
- **PlRpTlXtl** -- Instances of PlRpTlXtl are planning timeline displays.

- **PlRpTlZmBar** -- Instances of PlRpTlZmBar are zoom bars which know how to pop up a plan interval window.

library plsc

/ecs/formal/PDPS/PLS/PLANG/src/PWB/sc/src

Schedulable Resources.

The Schedulable resources, in fact, perform the scheduling of the activities on resources. For each resouce, there must be a corresponding schedulable resource with it.

Classes

- **PlScAllImpct** -- Instances of PlScAllImpct add the knowledge of oversubscription to the impact scheduler.
- **PlScComputer** -- Instances of PlScComputer are schedulable computer resources.
- **PlScCpu** -- Instances of PlScCpu are schedulable CPU resources.
- **PlScForceLink** -- Instances of PlScForceLink are used to force linkage of schedule resource model classes which may not be present at link time.
- **PlScLocalDisk** -- Instances of PlScLocalDisk are schedulable local disk resources.
- **PlScResource** -- This class sets up protocol for all ECS schedulable resources.

library PlSSIT

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library manages the Science Software Integration & Test PDPS DB interfaces.

Classes

- **PlAlternateCollection** -- It allow for all PlAlternate database records associated with a given PgeId to be retrieved from the database.
- **PlDataTypeFiles** -- This class captures the attributes that describe the relationship between Names of File Types (within a granule) and the integer value of that file type used by PDPS.
- **PlDataTypeFilesCollection**
- **PlDataTypeReqCollection**
- **PlDbIfManager** -- This file defines a helper class to manage the various database interfaces.
- **PlExitDependency**
- **PlExitDependencyCollection**
- **PlExitMessage**
- **PlExitMessageCollection**
- **PlMetaDataChecksCollection**
- **PlOutputYieldCollection**

library pltl

/ecs/formal/PDPS/PLS/PLANG/src/PWB/tl/src

Timeline.

This library contains classes responsible for buiding the Timeline application which helps visualize the scheduled activities in a particular plan. Almost all the classes are for building GUI, except the application and noid classes.

Classes

- **PlTlApDefsCfg** -- Instances of PlTlApDefsCfg handle configuration files for default display resource settings for the timeline.
- **PlTlAppl** -- An PlTlAppl is the main application class for the timeline.
- **PlTlColChgList** -- Instances of PlTlColChgList are lists which are part of the color mapper dialog.
- **PlTlColChgSt** -- Instances of PlTlColChgSt map colors to types.
- **PlTlColChngr** -- Instances of PlTlColChngr can be used to specify colors for types.
- **PlTlConfig** -- PlTlConfig is the base class for all configuration files.
- **PlTlDefsConfig** -- Instances of PlTlDefsConfig load and save basic timeline configuration files.
- **PlTlFileDiag** -- Instances of PlTlFileDiag are base classes dialogs for handling the loading and saving of configuration files.
- **PlTlFileSelDiag** -- This is a derived DFileSelectBox that creates a FileSelectionDialog (i.e. both the FileSelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make sure the parent
- **PlTlLoadDiag** -- Instances of PlTlLoadDiag are dialogs for loading saved timeline configuration files.
- **PlTlMainRegion** -- A PlTlMainRegion is a _TLMainRegion_ that implements "timeline" mouse tracking behavior.
- **PlTlPanel** -- This class, which is derived from TLPanel, adds capabilities for interactively changing colors and resources, and for saving and loading from different configuration files.
- **PlTlPlanSelDiag** -- This is a derived DSelectionBox that creates a PlanSelectionDialog (i.e. both the SelectionBox widget and its DialogShell parent are created no matter what create function you call.) In addition, this class adds behavior to make sure the DialogShell
- **PlTlPlanWinChgr** -- Instances of PlTlPlanWinChgr are dialogs for selecting a plan.
- **PlTlRsConfig** -- Instances of PlTlRsConfig are responsible for loading a set of names of resources to be displayed on the timeline.
- **PlTlRsEditor** -- Instances of PlTlRsEditor are dialogs for choosing the set of resources to be displayed on the timeline.

- **PlTlSaveDiag** -- Instances of PlTlSaveDiag are dialogs for saving timeline configuration files.
- **PlTlSimpleEvent** -- Instances of PlTlSimpleEvent provide base class behavior that all timeline simple events will provide.
- **PlTlSliderPup** -- Instances of PlTlSliderPup are popup windows which display a start time and stop time for the corresponding plan region.
- **PlTlSrmNoid** -- Instances of PlTlSrmNoid are timeline resource model client noids.
- **PlTlStatusBar** -- Instances of PlTlStatusBar provide a line at the bottom of the timeline for displaying messages.
- **PlTlStplConfig** -- Instances of PlTlStplConfig handle stipple keywords in configuration files.
- **PlTlStplEv** -- Instances of PlTlStplEv are stipple events.
- **PlTlStplEvent** -- Instances of PlTlStplEvent are stipple events associated with resource states.
- **PlTlStplView** -- Instances of PlTlStplView are views associated with stipple events.
- **PlTlStplVw** -- Instances of PlTlStplVw are views of stipple events.
- **PlTlTypesConfig** -- Instances of PlTlTypesConfig handle configureation files which can contain class names which are added to the list of the timeline's known types..
- **PlTlValEvent** -- This class is a derived TLEvent which is used for displaying values versus time data graphically on a TLEventSubRegion over a specific interval of time.
- **PlTlValView** -- This class is the base class for views that display value versus time information graphically in a timeline subregion.
- **PlTlWindow** -- PlTlWindow are main windows used by the PlTlAppl class.
- **PlTlXtl** -- Instances of PlTlXtl are planning timeline displays.
- **PlTlZmBar** -- Instances of PlTlZmBar are zoom bars which know how to pop up a plan interval window.

library PlUtil

/ecs/formal/PDPS/PLS/PLANG/src/CoreLib

This library contains utilities dealing with current time, time periods, time boundaries, time windows, time intervals, spacecraft, orbit, instruments, and also manages their DB interfaces.

Classes

- **PlBoundary** -- This class is used to describe boundaries for data acquisition and data processing within PGEs.
- **PlDbIfManager** -- This file defines a helper class to manage the various database interfaces.
- **PlInterval**

- **PlOrbitBoundary** -- The goal of this class is to encapsulate the knowledge of how to determine boundary times.
- **PlOrbitModel** -- The PlOrbitModel class is a table used by PlOrbitScheduled for calculating the estimated times for an orbit.
- **PlOrbitPeriod** -- The goal of this class is to store the platform of in a period such that the PlTime class can perform Period & Boundary operations.
- **PlPathModel** -- This class will give us one to one mapping of our current absolute path number to instrument-specific path number.
- **PlPeriod** -- This class defines a period of time.
- **PlTime** -- This class is a utility class for management of Time within the PDPS, it is derived from the standard Rogue Wave class but overrides it's behaviour to expect as a standard to be constructed from UTC, and to format to UTC, which is the standard that is
- **PlTimeService** -- Helper class to return ensure that only one instance of the EcTiTimeService class is initialized, so that when an application runs in the DCE environment it can use DCE time.

Templates

- **PlObjectPool**

library plwb

/ecs/formal/PDPS/PLS/PLANG/src/PWB/wb/src

Planning Workbench.

This library contains classes responsible for building the user interface of planning Workbench. These classes provide the users with the capability of creating candidate plans and schedule different activities (production requests) on them.

Classes

- **PlWbAppl** -- Instances of PlWbAppl are derived PlDiAppl's capable of using it's noid to request resources, plan names and activities from the resource model.
- **PlWbCatalog** -- The PlWbCatalog is a local process update catalog for updating the Scheduling System Interface pools.
- **PlWbMotifNotifier** -- Instances of PlWbMotifNotifier are Workbench specific Motif Event Notifiers which override the startPoll behavior.
- **PlWbNoid** -- Instances of PlWbNoids handle planning workbench related client noid messages.
- **PlWbOppGen** -- Instances of PlWbOppGen are opportunity generators for the production planning workbench.
- **PlWbPlanWin** -- A PlWbPlanWin is a dialog which allows selection of a plan.

- **PlWbPriorityWin** -- A PlWbPriorityWin is a window which allows selection of a priority producion requests.
- **PlWbQuestionWin** -- Instances of PlWbQuestionWin are popup windows for presenting questions to user/operator, and taking answers.
- **PlWbScheduler** -- The PlWbScheduler is the scheduler for the production planning workbench.
- **PlWbTimer** -- Instances of PlWbTimer are timers used for scheduling timeouts and control.
- **PlWbWin** -- This class represents the main Motif Window for the production planning workbench.

library RequestProcessing

/ecs/formal/DM/GatewayCSCI/src/RequestProcessing

Classes

- **AcquireItem** -- This class combines the disttribution parameters and the UR for each acquire to be submitted to the data server.
- **DmGwAcquireRequest** -- This Class contains all the information and the operations required to submit a product order request to the data server.
- **DmGwBrowseRequest** -- This Class contains all the information and the operations required to submit a Browse request to the data server.
- **DmGwDirectoryRequest** -- This class performs abort, submit, extrac header, and build output information for the directory request.
- **DmGwErrorResult** -- This class performs build error and return error.
- **DmGwInvESDTReference** -- This class performs get spatial extent, rectangle, point, polygon, start & end date, UR.
- **DmGwInvQuery** -- This class performs creating gateway inventory query object.
- **DmGwInvRequests** -- This class performs submit, abort, find data servers, extrac search informations, build inventory queries, build search requests, and build the outputs.
- **DmGwInvSearchRequest** -- This class performs submit, abort, complete, failed, and connect data server.
- **DmGwL7AcquireRequest** -- This Class contains all the information and the operations required to submit a product ordering request to the LIM .
- **DmGwODLLabelIO** -- DmGwODLLabelIO class provides threadsafe interface to read/write the ASCII ODL definitions from/to ODL tree.
- **DmGwProductRequest** -- This class extracts the Product Request ODL message, processes the request and formulates the result ODL message for the client.
- **DmGwSearchRequest** -- This class contains all the information and the operations required to submit a search request to the data server for production history or ancillary data UR.

- **DmGwV0BrowseRequest** -- This class extract the Browse Request parameters from the ODL tree, submits the requests.
- **DmGwV0ECSMapper** -- This class maps the V0 types to ECS types and vice versa.
- **LineItem** -- This class represents the V0 LINEITEM type in Product Request message.
- **ServerAlloc** -- This class is used for indicate its UUID and if the server is busy.

library Server

/ecs/formal/DM/GatewayCSCI/src/Server

library srf

/ecs/formal/COMMON/CSCI_SRF/src/srf

The SRF library provides the necessary mechansim for the application to communicate asynchronously with another application.

Classes

- **EcSrAsynchRequest_C** -- This class represents the context of an SRF request currently in execution.
- **EcSrAsynchRequest_S** -- This is the server side representation of a request in progress.
- **EcSrIDTime** -- The only purpose of this class is to put the structure of reqID, the time when the request is put in the dictionary, and the timeout in to a list that allows the cleanupThread to index through it.
- **EcSrRequestDispatcher** -- This class performs request dispatching.
- **EcSrRequestServer_C** -- This class represents the clients view of an SRF server.
- **EcSrRequestServer_S** -- This is the server side representation of a client/server relationship.
- **EcSrTransport_1_0** -- This is the transport server for SRF.
- **EcSrTransport_1_0_ABS** -- Abstract base class for EcSrTransport.
- **EcSrTransport_1_0_Mgr** -- Manager class for EcSrTransport.

library srfSh (shared)

/ecs/formal/COMMON/CSCI_SRF/src/srf

The SRF library provides the necessary mechansim for the application to communicate asynchronously with another application.

Classes

- **EcSrAsynchRequest_C** -- This class represents the context of an SRF request currently in execution.
- **EcSrAsynchRequest_S** -- This is the server side representation of a request in progress.

- **EcSrIDTime** -- The only purpose of this class is to put the structure of reqID, the time when the request is put in the dictionary, and the timeout in to a list that allows the cleanupThread to index through it.
- **EcSrRequestDispatcher** -- This class performs request dispatching.
- **EcSrRequestServer_C** -- This class represents the clients view of an SRF server.
- **EcSrRequestServer_S** -- This is the server side representation of a client/server relationship.
- **EcSrTransport_1_0** -- This is the transport server for SRF.
- **EcSrTransport_1_0_ABS** -- Abstract base class for EcSrTransport.
- **EcSrTransport_1_0_Mgr** -- Manager class for EcSrTransport.

library UIComponent

/ecs/formal/CLS/DESKT/src/MotifApp

This library contains generic user interface components used by the desktop such as menu bars, busy cursors, etc.

Classes

- **Application** -- Allow main and MainWindow to access protected member functions.
- **AskFirstCmd** -- Base class for Cmds that ask for confirmation.
- **BasicComponent** -- First version of a class to define a protocol for all components.
- **BusyPixmap**
- **ButtonInterface** -- A pushbutton interface to a Cmd object.
- **Clock**
- **Cmd** -- A base class for all command objects.
- **CmdInterface**
- **CmdList** -- Maintain a list of Cmd objects.
- **ColorChooser**
- **ColorModel** -- An RGB color model for a single color.
- **DialogManager** -- A base class for cached dialogs.
- **HSVView** -- Display the contents of a ColorModel as HSV color components.
- **IconifyCmd** -- Iconify all windows in a MotifApp application.
- **InfoDialogManager**
- **MainWindow** -- Support a top-level window.
- **ManageCmd** -- Manage all windows in a MotifApp application.
- **MenuBar** -- A menubar that supports panes of Cmd objects.
- **MenuWindow** -- Add a menubar to the features of MainWindow.

- **NoUndoCmd** -- Base class for all commands without undo.
- **PixmapCycler** -- Class that supports a continuous cycle of pixmaps for short animation sequences.
- **QuestionDialogManager**
- **QuitCmd** -- Exit an application after checking with user.
- **RGBController** -- Control the ColorModel.
- **RGBView** -- Display the contents of a ColorModel as RGB color components.
- **SelectFileCmd** -- Allow the user to select a file interactively.
- **SwatchView** -- Display a color "swatch" corresponding to the color in the ColorModel.
- **TextView** -- Base class for all text (numerical) views of a ColorModel.
- **UIComponent** -- Base class for all C++/Motif UI components.
- **UndoCmd** -- An interface for undoing the last command.
- **WarnNoUndoCmd** -- Warns user before executing a command.
- **WorkingDialogManager**

Templates

- **SimpleList** -- A Simple List Template.

library xpm

/ecs/formal/CLS/DESKT/src/libxpm

The Xpm library is a public domain pixmap display library. It provides functions to read and write pixmap format and is used by the Desktop to display the icons associated with files.

This page intentionally left blank.

Abbreviations and Acronyms€

ABC++	Document Generator used to provide class level detail
ACFG	Management Agent configuration file
ACL	Access Control List
ACMHW	Access and Control Management Hardware (Configuration Item)
ADC	Affiliated Data Center (National Oceanic and Atmospheric Administration only)
ADSHW	Advertising Service Hardware (Configuration Item)
ADSRV	Advertising Service (Computer Software Configuration Item)
AIT	Algorithm Integration and Test
AITHW	Algorithm Integration and Test Hardware (Configuration Item)
AI&T	Algorithm Integration and Test
AITTL	Algorithm Integration and Test Tools (Computer Software Configuration Item)
AM-1€	EOS AM Mission spacecraft 1, morning equator crossing spacecraft series -- ASTER, CERES, MISR, MODIS and MOPITT instruments
AMASS	Archival Management and Storage System *Storage Management Software created by EMASS, which is the Raytheon Company, which in turn owns a part of Hughes Aircraft company, owns a wholly owned subsidiary of E-Systems Inc. E-Systems.* *For more info you can visit: http://www.emass.com*
AOS	ASTER Operations Segment
AP	Algorithm Package
APC	Access/Process Coordinators
API	Application Program Interface
AQA	Algorithm Quality Assurance
AQAHW	Algorithm Quality Assurance Hardware (Configuration Item)
AR	Action Request
ARS	Remedy Action Request System

AS	Administration Stations
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
ATM	Asynchronous Transfer Mode
BBS€	Bulletin Board Server Bulletin Board Service Bulletin Board System
BCP	Bulk Copy Program
BDS	Bulk Data Service
BLM	Baseline Manager
B0SOT	B.0 Search and Order Tool
CAD	Computer Aided Design
CCDI	ClearCase DDTS Integration
CCR	Configuration Change Request
CDE	Common Desktop Environment
CD-ROM	Compact Disk - Read Only Memory
CDS	Cell Directory Service
CERES	Clouds and Earth’s Radiant Energy System
CGI	Common Gateway Interface
CHUI	Character-based User Interface
CI	Configuration Item
CLS	Client Subsystem
CM	Configuration Management
CMI	Cryptographic Management Interface
COTS	Commercial Off the Shelf (Software or Hardware)
CPU	Central Processing Unit
CRM	Change Request Manager
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSMS	Communications and Systems Management Segment (ECS)
CSS	Communications Subsystem
DAA	Data Availability Acknowledgement

DAAC	Distributed Active Archive Center
DAN	Data Availability Notice
DAP	Delivered Algorithm Package
DAR	Data Acquisition Request
DAS	Dual Attached Station
DB	Database
DBMS	Database Management System
DCCI	Distributed Computing Configuration Item
DCE	Distributed Computing Environment (OSF)
DDA	Data Delivery Acknowledgment
DDICT	Data Dictionary (Computer Software Configuration Item)
DDIST	Data Distribution (Computer Software Configuration Item)
DDN	Data Delivery Notice
DDT	DAAC Distribution Technician
DDTS	Distributed Defect Tracking System
DESKT	Desktop (Computer Software Configuration Item)
DEV	Custom Developed Code
DFS	Distributed File System
DID	Data Item Description
DIPHW	Distribution and Ingest Peripheral Hardware Configuration Item
DLL	Dynamic Link Library
DLT	Digital Linear Tape
DMGHW	Data Management Hardware (Configuration Item)
DMS	Data Management Subsystem
DNS	Domain Name Service
DOF	Distributed Object Framework
DPR	Data Processing Request
DPRID	Data Processing Request Identifier
DPS	Data Processing Subsystem
DR	Data Repository

DRPHW	Data Repository Hardware (Configuration Item)
DSS	Data Server Subsystem
DTS	Distributed Time Service
EBnet	Earth Observing System Data and Information System Backbone Network
ECS	Earth Observing System Data and Information Core System
EDC	Earth Resource Observation System Data Center
EDN	Expedited Data Set Notification
EDOS	Earth Observing Data and Operations System
EDR	Expedited Data Set Request
EDS	Expedited Data Set
EISA	Enhanced Industry Standard Architecture
E-mail	Electronic Mail (also Email, e-mail, and email)
EMASS	E-Systems Modular Automated Storage Systems
EOC	Earth Observing System Operations Center
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EROS	Earth Resource Observation System
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth Science Data Type
ESOD	Earth Science Online Directory
ESRI	Environmental Systems Research Institute
ETM+	Enhanced Thematic Mapper Plus
FCAPS	Fault, Configuration, Accountability, Performance, and Security
FDDI	Fiber Distributed Data Interface
FH	Fault Handling
FLDB	Fileset Location Database
F&PRS	Functional and Performance Requirements Specification
FSMS	File and Storage Management System
FTP	File Transfer Protocol

FTPD	File Transfer Protocol Daemon
GB	gigabyte (10^9)
GDPS	Ground Data Processing System
GDS	Ground Data System
GFE	Government Furnished Equipment
GSFC	GODDARD Space Flight Center
GSMS	Ground System Management Subsystem (ASTER)
GTWAY€	(Version 0 Interoperability) Gateway (Computer Software Configuration Item)
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HIPPI	High Performance Parallel Interface
HMI	Human Machine Interface
HP	Hewlett Packard
HPOV	Hewlett Packard Open View (COTS product)
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
HWCI	Hardware Configuration Item
IAS	Image Assessment System
IBM	International Business Machines
ICD	Interface Control Document
ICLHW	Ingest Client Hardware (Configuration Item)
ID	User Identification (or Identifier)
IDL	Interactive Data Language
I/F	Interface
ILG	Infrastructure Library Group
ILM	Inventory, Logistics, Maintenance Manager
IMS	Information Management System
INGST	Ingest (Computer Software Configuration Item)
INS	Ingest Subsystem

I/O	Input/Output
IOS	Interoperability Subsystem
IP	Installation Plan International Partner Internet Protocol
IRD	Interface Requirements Document
ISO	International Standards Organization
ISS	Internetworking Subsystem
I&T	Integration and Test
JEST	Java Earth Science Tool
KFTP	Kerberos File Transfer Protocol
L0 - L4	Level-0 through Level-4 data
L0R	Landsat Reformatted Data
LAN	Local Area Network
Landsat-7	Land Remote-Sensing Satellite
LaRC	Langley Research Center
LaTIS	LaRC TRMM Information System
LFS	Local File System
LIM	Local Information Manager
LIMGR	Local Information Manager (Computer Software Configuration Item)
LPS	Landsat-7 Processing System
LSM	Local System Management
LZ77	Lampel-Ziv coding
MACI	Management Agent Configuration Item (Computer Software Configuration Item)
MB	Megabyte (10^6)
Mbps	Megabits Per Second
MCF	Metadata Configuration File
MCI	Management Software Configuration Item (Computer Software Configuration Item)

MDA	Management Data Access Software Component
MHWCI	(System) Management (Subsystem) Hardware Configuration Item
MIB	Management Information Base
MISR	Multi-Imaging SpectroRadiometer
MLCI	Management Logistics Configuration Item (Computer Software Configuration Item)
MM	Mode Management
MMO	Mission Management Office
MMS	Mode Management Service
M&O	Maintenance and Operations
MOC	Mission Operations Center
MODIS	Moderate-Resolution Imaging SpectroRadiometer
MOJO	Message Oriented JEST Objects
MOPITT	Measurements of Pollution in the Troposphere
MP	Message Passing
MS	Mass Storage Microsoft
MSS	System Management Subsystem
NASA	National Aeronautics and Space Administration
NCR	Non-conformance Report
NFS	Network File System
NIS	Network Information Service
NMC	National Meteorological Center (located at National Oceanic and Atmospheric Administration)
NNM	Network Node Manager
NNTP	Network News Transfer Protocol
NOAA	National Oceanic and Atmospheric Administration
NSI	National Aeronautics and Space Administration Science Internet
NSIDC	National Snow and Ice Data Center
ODL	Object Description Language

OEM	Original Equipment Manufacturer
OODCE	Object Oriented DCE
OS	Operating System
OSI	Open Systems Interconnect
OV	Open View
OVW	Open View for Windows
PC	Personal Computer
PCM	Physical Configuration Management
PDPS	Planning and Data Processing Subsystems
PF	Process Framework
PGE	Product Generation Executive
PLANG	Production Planning (Computer Software Configuration Item)
PLNHW	Planning Hardware (Configuration Item)
PLS	Planning Subsystem
PNM	Physical Network Manager
PRONG	(Data) Processing (Computer Software Configuration Item)
QA	Quality Assurance
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RCS	Request Communications Support
RDBMS	Relational Database Management System
RFA	Remote File Access
RFC	Request For Comments
RIP	Routing Information Protocol
RPC	Remote Procedure Call
SAGE III	Stratospheric Aerosol and Gas Experiment III
SAS	Single Attached Station
SATAN	Security Administrator Tool for Analyzing Networks
SBSRV	Subscription Server
SCF	Science Computing Facility

SCSI	Small Computer System Interface
SDP€	Science Data Plan
	Science Data Processing
	Software Development Plan
SDPS	Science Data Processing Segment (ECS)
SDPTK	Science Data Processing Toolkit
	Science Data Processing Toolkit (Computer Software Configuration Item)
SDSRV	Science Data Server Computer Software Configuration Item
SGI	Silicon Graphics, Inc.
SMC	System Management Center
SMP	Symmetric Multi-processor
SMUX	Simple Network Management Protocol Multiplexing
SNMP	Simple Network Management Protocol
SPRHW	Science Processing Hardware (Configuration Item)
SQL	Structured Query Language
SQS	Spatial Query Server
SRF	Server Request Framework
SSAP	Science Software Archive Package
SSIT	Science Software Integration and Test
STK	StorageTek
STMGT	Storage Management (Computer Software Configuration Item)
SYSLOG	System Log
TAR	Tape Archive
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
T/EC	Tivoli/Enterprise Console
TMR	TOPEX/Poseidon Microwave Radiometer
TT	Trouble Ticket
UDP	User Datagram Protocol
UR	Universal Reference

URT	User Registration Tool
UUID	Universal Unique Identifier
V0	Version Zero
V0 GTWAY	Version Zero Gateway
VFS	Virtual File Service
VIP	Versatile Interface Processor
VOB	Version Object Base
VT	Virtual terminal
WKBCH	Workbench (Computer Software Configuration Item)
WKSHW	Working Storage Hardware Configuration Item
WS	Working Storage
WWW	World Wide Web
xAR	x Acquisition Request (where x is a generic acquisition request)
XBDS	Bulk Data Service Protocol
XDR	External Data Representation
XFS	Extended File System
XVT	XVT Software, Inc.
Zmail	UNIX mail tool